FM77AV

VISUAL入門

岳田陽介 著



工学図書株式会社

FM77AV VISUAL入門

岳田陽介 著

工学図書株式会社

はじめに

パーソナル・コンピュータが世に出始めてから，まだ20年を経ていない。始めはオモチャ程度の機能しかなかったのが，今や成人になろうとしている。この間に全く目ざましいほどの発達を遂げたものである。十数年前まではヨチヨチ歩きだったのが，今では，すっかり社会に定着し，ビジネスの世界ではなくてはならないものにまで成長してきた。

もう一つ，無くてはならなくなった世界がある。それが子供の世界。ゲームの世界である。

現在ある種のゲーム専用パソコンが600万台売れているという。ファミコンがそれだ。ファミコンだって，レッキとしたコンピュータである。ただ低兼化を目指しているため，ゲーム専用機として作られており，ゲーム用の機能しか備えていないという違いがあることは事実である。しかし，子供達はこのファミコンに触れ，ファミコンを使いこなすことにより，コンピュータを道具として受入れるという意識が，しだいに定着していることは事実である。

今ファミコンを大人がとうてい及ばない程の驚くべき能力を発揮して，自由に駆使している彼等は，明日の本格的コンピュータのユーザである。彼等がファミコンを通して，コンピュータを理解し，コンピュータと話することを知り，コンピュータの能力と限界を体得し，自分の身体の一部のようにとり込んでいる様子をみていると，すでに，新しい文化が生れつつあるように思える。これまで，支えてきた，物質文明から，コンピュータを通した情報文化の入口を見ているような気がする。

このようなコンピュータ文化のなかで，自分も使う側からもう少し，作る側に近づいて，より密接にコンピュータに触れようとする人々に，趣味の世界からアプローチしょうとするのが，この本の試みである。幸にして，いまパソコンは，美しい画面

を手軽につくり出すことができるように高機能化してきた。この機能を十二分に駆使して，楽しみながら，しだいにコンピュータの魅力に溶け込んでゆくことができればと，そんな願いを込めてこの本を書いてみた。

プログラムを入力し，それが一つ一つ動き出すことを確認するたびに，あたかも，コンピュータに命を吹き込むような感激を覚えることがある。この喜びを読者諸氏と共に味わうことができれば，著者にとってこれ以上の喜びはない。

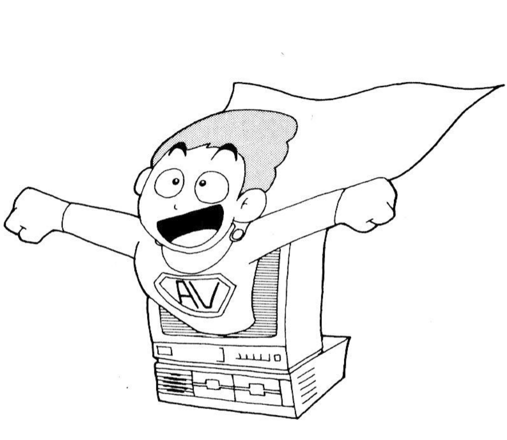
本書が，これから加速されるコンピュータ化の波に同調するための足がかりをつけるキッカケとなってもらえれば幸いである。

昭和61年3月

著　者

もくじ

1 FM77AVとは



1.1 FM77AVの特徴

AVとは，Audio Visualを略したもので，音と画像を楽しむパソコンという感覚でつけた名前だ．

そのために，これまでの8ビット・マシーンでは考えられなかったような，数々の特色が生かされているのが面白い．

・4096色を表現できるグラフィック機能

320×200ドットでは4096色，640×200ドットでは8色（2画面分）の表示ができる．

・FM音源によるミュージック機能

FM音源を加えることにより，一度に6重和音が出せるようになり，シンセサイザーに近い機能を発揮できる．

・スーパーインポーズ機能

テレビの画面と，パソコンの画面をドッキングする機能．

これらの機能を組み合わせて，使うことにより，これまで描いていた，パソコンのイメージを大きく飛躍させることができるようになった．

単なる計算→作表機から，もっと視覚的なツールへ，単純なゲーム機から，より迫力のあるAVマシーンへと，楽しさは広がるばかりである．

本機を前にして，本機に触れながら，頭の中にある空想の世界を現実のものに表現してみようではないか．

1.2 FM77AVの楽しみ方

これまでパソコンを使ったことがない人でも，また使い慣れた人でも，もっと多くの楽しみ方を発見したいはずである．

一昔前，プラモデルのブームがあった．これまで，木片から刻り出した車や船とは段違いな程精密なモデルには，いつも子供の夢を大きくふくらませてくれたものであった．

それと同じように，パソコンも子供達の心をすっかりとらえてしまった．ゲームから始まり，より複雑なプログラムを，いとも簡単に操つる彼等を前にして，大人達には異星人を見るかのような驚きがある．

彼等は，より高度なプログラムへと次々にプログラムのレベルを上げ，新しい世界に挑戦していくのである．そんな彼等を含めて，より広い層に，パソコンの遊びの世界を紹介したい．

このパソコンは，幸いにして，豊富な画面表示機能を持っているので，本書では，特にグラフィックに重点を置いた，プログラムの作り方を紹介していくことにする．

ごく簡単な動き，表示法から始まり，複雑なゲームの作り方まで，例示してみよう．一つ一つ，自分で入力し，動かしてみると，新たな感動が伝わってくるはずである．

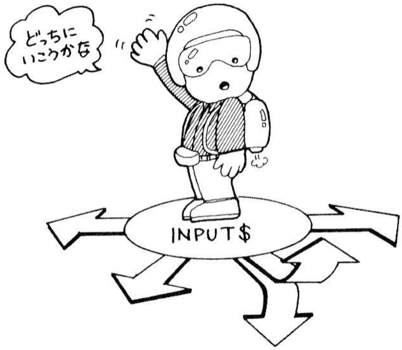
本書は，ゲームを通して，BASICの扱い方，画面の取扱い方，アイデアをプログラムに生かしていく方法について，その

一端を例示していくことにする.

カラフルな画面を体験し，トリッキーな画面に挑戦し，そして，自分でもっとオリジナルなプログラムを生み出す力をつけてもらいたい.

2

動きのハウ・ツー

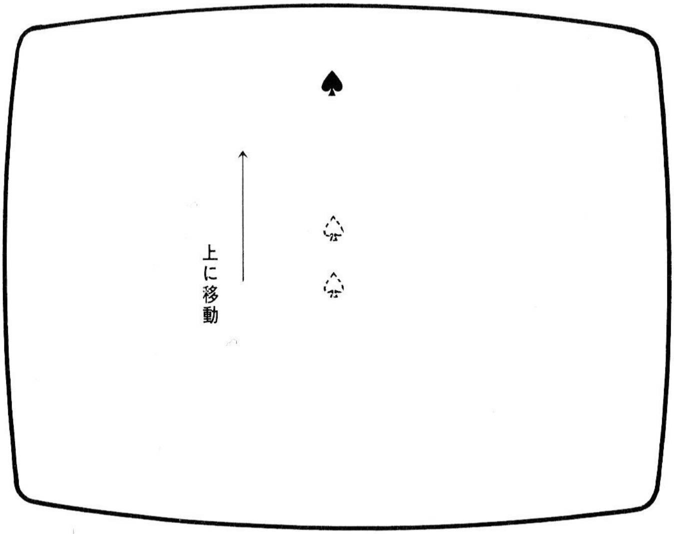


2.1 上への移動

はじめに，もっとも簡単な移動・表示の方法を体験してみよう．
シンボルで，スペードのマークを画面中央から上に移動しながら表示してみる．

右のプログラムを正確に入力して実行してみよう．行10から行30までは別に入力しなくてもよい．
行130から160までの前半のプログラムでは動いた軌跡が残るが，後半の行200から250までのプログラムは軌跡を表示しないようになっている．
ここで，ちょっと便利なワンポイント・プログラムの解説をしておこう．行180．これは，INPUT$文で，１文字どれかのキーが押されるまでプログラムはここで中断する．この命令は，１文字入力しても，画面に表示（エコーバック）しないし，リターン・キーを押さなくてもすむので，プログラムを一時停止させ，任意のキーで再スタートさせるのに便利に使えるので，覚えておこう．

```
10 '  P2.1  ｳｴ ﾍﾉ ｲﾄﾞｳ
20 '
30 '                                変数Yは表示行を示す
100 SCREEN@ 0:CLS
110 A$="♠": B$=" "
120 '
130 FOR Y=12 TO 0 STEP -1      ← 表示位置を1行ずつ上へずらす
140   LOCATE 40,Y : PRINT A$   ← スペードの表示
150     FOR T=1 TO 500:NEXT T  ← 移動スピード調節用のタイマー
160 NEXT Y
170 '
180   K$=INPUT$(1)             ← プログラムの一時中止用
190 '                            (任意のキーを押すと次へ進む)
200 CLS
210 FOR Y=12 TO 0 STEP -1
220   LOCATE 40,Y : PRINT A$
230     FOR T=1 TO 500:NEXT T
240   LOCATE 40,Y : PRINT B$   ← 前の表示結果を消す
250 NEXT Y
```



行200以後を実行した結果

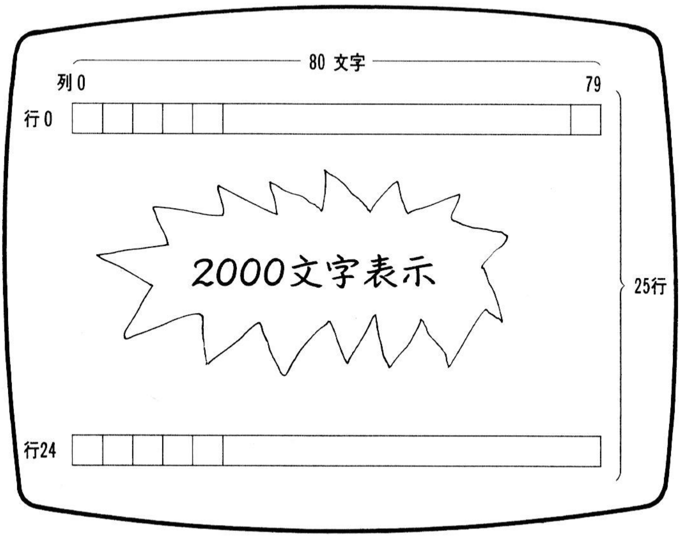
2.2 下への移動

前のプログラムＰ２．１をちょっと変えただけで，スペードのマークは上に移動せずに，下に移動する．行130，行210を次のように変えてみよう．

```
130 FOR Y=12 TO [0] STEP -1     130 FOR Y=12 TO 20
                 ↓ ~~~~~~~~~
                20    省略    ⇨
                 ↑ ~~~~~~~~~
210 FOR Y=12 TO [0] STEP -1     210 FOR Y=12 TO 20
```

Ｙはスペードマークを表示する縦の位置(何行目か)を示す値である．変更後の行130，210は，12行目から20行目まで，１行ずつ増加させなさい，(ということは，下に移動させながらということになる．) という指示になる．

```
10 '  P2.2  ｼﾀ ﾍﾉ ｲﾄﾞｳ
20 '
30 '
100 SCREEN@ 0:CLS
110 A$="♠": B$=" "
120 '
130 FOR Y=12 TO 20
140   LOCATE 40,Y : PRINT A$
150     FOR T=1 TO 500:NEXT T
160 NEXT Y
170 '
180   K$=INPUT$(1)
190 '
200 CLS
210 FOR Y=12 TO 20
220   LOCATE 40,Y : PRINT A$
230     FOR T=1 TO 500:NEXT T
240   LOCATE 40,Y : PRINT B$
250 NEXT Y
```



文字表示位置

2.3 左右の移動

上，下の動きを左，右に変えてみよう．左右に移動させるためには，表示位置を横に動くようにプログラム上で指定すればよい．今まで，上・下に動かすことを学んできた．そのプログラムで，キーポイントは，指定した位置に表示させるLOCATE文の使い方だ．

LOCATE X，Y

Y：縦方向の位置を指定する（0～24）

X：横方向の位置を指定する（0～79）

X，Yは縦，横方向の位置を指定するために仮りにつけた変数名．何でもかまわない．ただし，指定できる値はWIDTH文による最大表示範囲を越えてはならないという制限があるので，要注意．

例えば

```
100 LOCATE 39, 12
110 PRINT "A"
```

を実行すると，横40文字目，縦13行目（スクリーンのほぼ中央位置）に，英字のAが表示される．

このように，LOCATE文は，スクリーン上に表示する文字の位置を自由に指定することができるので，この文を効果的に使えば，文字を移動表示できることになる．

ここの例では，LOCATE 文の X 方向(横方向)の値を変えることによって，スペードマークを横に動く例を示している．X の値を増やすと右へ，逆に減らすと左へ移動する．

つぎのプログラム（P 2．3）は X を40から80へ増やしているので，スペード・マークは，スクリーンの中央位置から右へ動くことになる．

```
10 '  P2.3  ｻﾕｳ ﾉ ｲﾄﾞｳ
20 '
30 '
100 SCREEN@ 0:CLS
110 A$="♠": B$=" "
120 '
130 FOR X=40 TO 80
140   LOCATE X,12 : PRINT A$
150     FOR T=1 TO 100:NEXT T
160 NEXT X
170 '
180   K$=INPUT$(1)
190 '
200 CLS
210 FOR X=40 TO 80
220   LOCATE X,12 : PRINT A$
230     FOR T=1 TO 100:NEXT T
240   LOCATE X,12 : PRINT B$
250 NEXT X
```

逆に左へ動かすには，行130, 210をつぎのように変えるとよい．

```
130 FOR X=40 TO 0 STEP -1
```

```
210 FOR X=40 TO 0 STEP -1
```

2.4 斜の移動

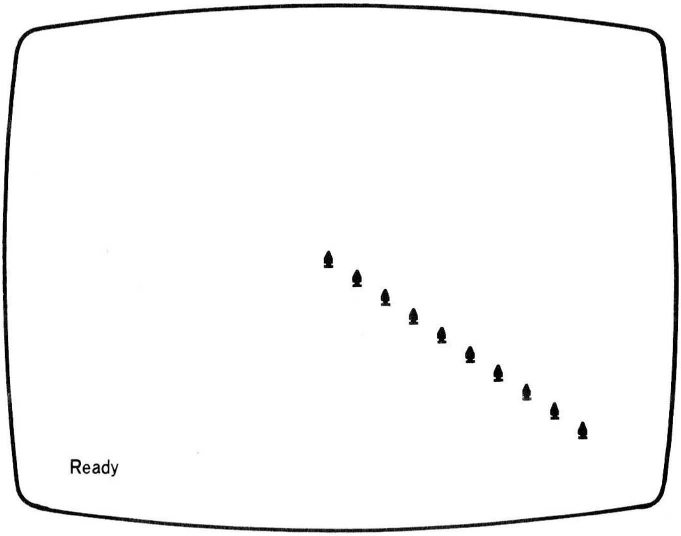
文字パターンを斜に移動させるためには，これまでの上・下，水平移動を同時に行えばよい．つぎのプログラムがその1例だ．

```
10 '   P2.4  ﾅﾅﾒ ﾉ ｲﾄﾞｳ
20 '
30 '
100 SCREEN@ 0:CLS
110 A$="♠": B$=" "
120 '
130 FOR N=1 TO 10
140   LOCATE 30+N*4,11+N : PRINT A$
150     FOR T=1 TO 500:NEXT T
160 NEXT N
170 '
180   K$=INPUT$(1)
190 '
200 CLS
210 FOR N=1 TO 10
220   LOCATE 30+N*4,11+N : PRINT A$
230     FOR T=1 TO 500:NEXT T
240   LOCATE 30+N*4,11+N : PRINT B$
250 NEXT N
```

N：パターンを10回表示するための変数

この例はスクリーンのほぼ中央位置から右下方へ，斜にパターンを移動表示するプログラムである．

X方向はY方向に比べ移動量が小さいので，動きを均等にするため，X方向の移動量を4倍にしていることに注意．



プログラムP2.4の行100—160までを実行した例

2.5 円運動

斜に移動させるのを，もう少し工夫すると，円運動をさせることができる．つぎの例で，点を表示する PSET 文を用いて，黄色の点で円を描くプログラムを示す．

```
10 '  P2.5  ｴﾝ ｳﾝﾄﾞｳ
20 '
30 '
100 SCREEN@ 0:CLS
120 '
130 FOR TH=0 TO 6.28 STEP .1
140   X=320+150*COS(TH)
150   Y=100+60*SIN(TH)
160 PSET(X,Y,6)
180     FOR T=1 TO 50:NEXT T
200 NEXT TH
```

この例では，SCREEN@ 0で指定する 8 色モード (640×200 ドット) の場合で実行してみる．

X, Y の初めの値として，それぞれ，320と100を与え，円の中心位置を，スクリーンのほぼ中心に設定する．

円を真円に近くするため，X 方向，Y 方向の径の長さを，それぞれ150と60にし，見やすい円形になるように調節．この値を変えると楕円形になる．

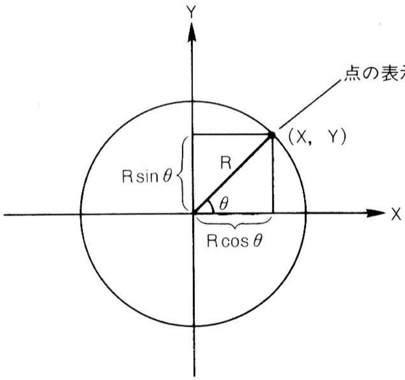
角度 θ に相当する変数は，弧度法（ラジアン）を用いているため，0 ～ 2 π の値をとる．

以上を実行すると，つぎのように，スクリーンの中心に，黄色の点で，円形が表示されることになる．

PSET 文はつぎのように使う.

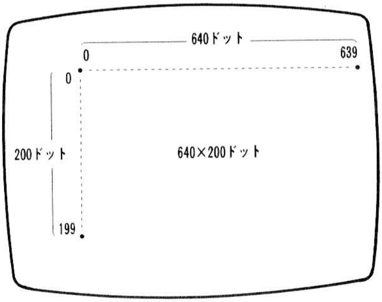
```
PSET (X, Y, カラーコード, 表示機能)
      └─┬─┘                  ↑
        ↑                    └── 省略可能
        └── 点の表示位置
```

ここで円運動をさせるのに，点の表示位置 X，Y を求めるのに，三角関数を用いてみた.

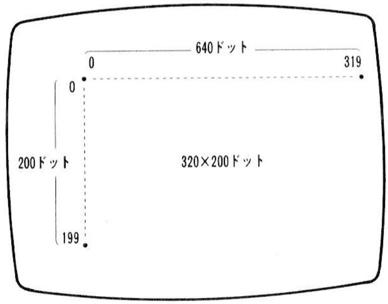


これまで，LOCATE 文で指定する文字の位置は，X 方向が 0 ～79，Y 方向が 0 ～24であった.

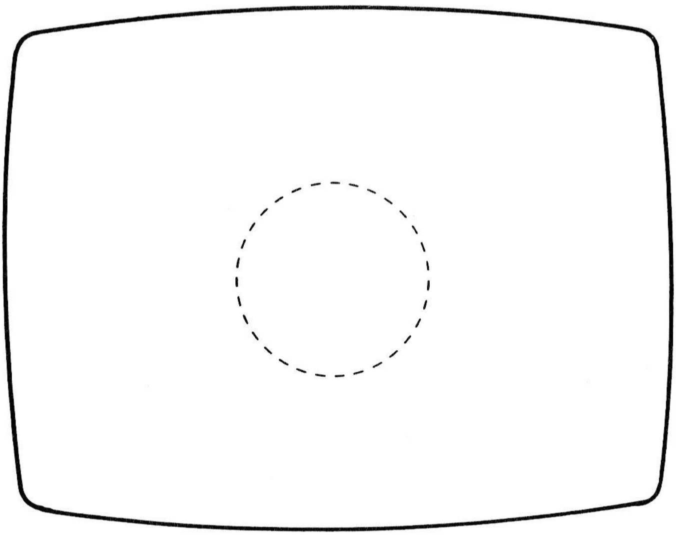
しかし，PSET 文で指定する点の表示位置は，グラフィック座標となり，つぎの広さを持つ.



SCREEN@0で指定する8色モードの場合



SCREEN@1で指定する4096色モードの場合



3 より複雑に動かす



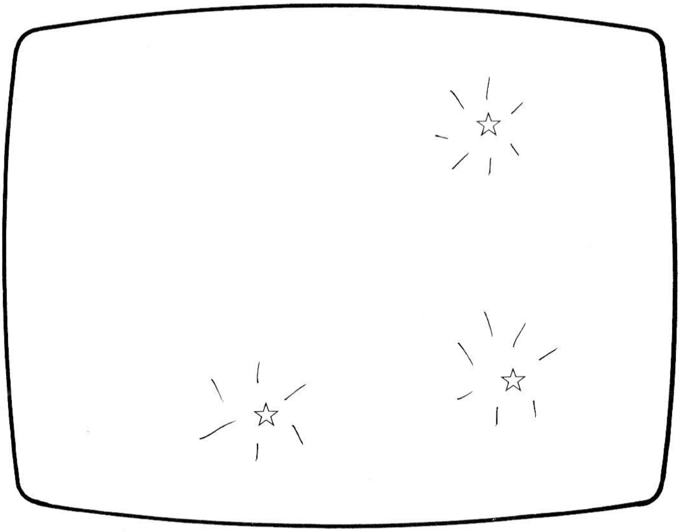
3.1 ランダムに表示

ここでは，画面上のランダム（不特定）な場所に "＊" のマークを表示してみよう．そのためには，まずランダムな場所を求める命令を知らなければならない．

X＝RND ＊79

このような文を実行すると，X には0～79までのでたらめな数が割りあてられる．これを X と Y について，画面のとり得る値を最大値となるように，行150のごとく実行すると，画面上の任意の場所 X 列，Y 行に "＊" が表示されることになる．

```
10 '  P3.1  ﾗﾝﾀﾞﾑ ﾋｮｳｼﾞ
20 '
30 '
100 SCREEN@ 0:COLOR 2 : CLS
110 A$="*": B$=" "
120 LINE(0,0)-(639,199),PSET,6,B
130 '
140 FOR N=1 TO 40
150   X=RND*78+1 : Y=RND*22+1
160   LOCATE X,Y : PRINT A$
170     FOR T=1 TO 100 :NEXT T
180   LOCATE X,Y : PRINT B$
190 NEXT N
200 COLOR 7 : CLS
```



3.2 ランダム・ウォーク

スクリーンいっぱいに，ヨッパライがフラフラ歩き回る様子を表示してみよう．

```
10 '  P3.2  ﾗﾝﾀﾞﾑ ｳｫｰｸ
20 '
30 '
100 SCREEN@ 0:COLOR 5 : CLS
110 A$="♣": B$=" "
120 X=40 : Y=12
130 LINE(0,0)-(639,199),PSET,6,B
140 '
150 FOR N=1 TO 200
160   R=INT(RND*4)+1                         ┐← 上，下，左，右のどっち
170   ON R GOSUB 270,280,290,300             ┘  に進むかランダムに選択
180     IF X<=0 THEN X=X+1:GOTO 160          ┐
185     IF X>=79 THEN X=X-1:GOTO 160         │
190     IF Y<=0 THEN Y=Y+1:GOTO 160          ├← 境界を越えないよ
195     IF Y>=24 THEN Y=Y-1:GOTO 160         ┘  うに規正
200     LOCATE X,Y : PRINT A$
210       FOR T=1 TO 100:NEXT T
220     LOCATE X,Y : PRINT B$
230 NEXT N
240 COLOR 7 : CLS
250 END
260 '
270 X=X+1 : RETURN   ┐
280 X=X-1 : RETURN   │
290 Y=Y+1 : RETURN   ├ ヨロケル歩幅を決める
300 Y=Y-1 : RETURN   ┘
```

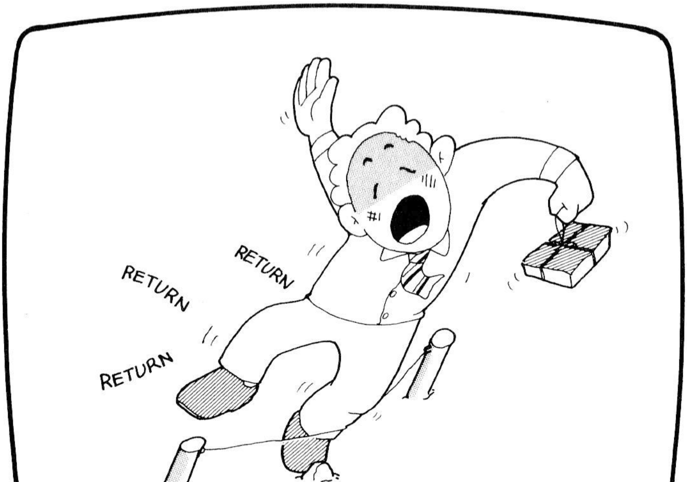
このプログラムを実行すると，ヨッパライ役のクローバ・マークが，ヒョロヒョロと歩き回る．

このプログラムのポイントは2つある．

第１：１～４の乱数を行160で発生させ，行170の GOSUB 文で，選ばれた乱数に対応し，上，下，左，右へ動くためのサブルーチン行270～行300に分岐させているところ．

第２：行180～行195までのところで，境界を越えそうになったら，引っぱりもどしているところ．

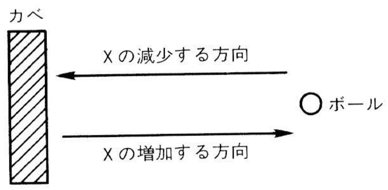
かくして，ヨッパライ君は，ドブにも落ちずに，フラフラと無事（？）に画面上を散索できることになる．



3.3 左右に反射

ボールを左右に反射させるにはどうすればよいのだろう．テニス・ゲーム，ブロックくずしなどのゲームは，このボールの移動，反射の動きを使っている．一見難かしそうな動きも，すべてこれまでのテクニックを応用すればよいことがわかる．

ボールの移動はＸ方向の値を変化させ，反射して，方向を変えるのは，Ｘの値を増加させるのか，減少させるのかの切り変えをすることにより達成できる．



```
10 '  P3.3   ﾊﾝｼｬ ｳﾝﾄﾞｳ
20 '                                  D ：進む方向を示す変数
30 '                                  DT：ボールの進むスピードを示す変数
100 SCREEN@ 0:COLOR 5 : CLS
110 A$="○": B$=" "
120 X=40 : Y=12 : D=-1: DT=200
130 LINE(0,75)-(7,125),PSET,4,BF      } 右左のカベの表示
140 LINE(632,75)-(639,125),PSET,4,BF  }
150 '
160 FOR N=1 TO 500
170   X=X+D
180     IF X<=1 THEN GOSUB 270    ← 左のカベにぶつかれば行270へ
190     IF X>=78 THEN GOSUB 270   ← 右のカベにぶつかれば行270へ
200     LOCATE X,Y : PRINT A$
210       FOR T=1 TO DT:NEXT T    ← しだいに速く移動させるための
220     LOCATE X,Y : PRINT B$        タイマー
230 NEXT N
240 COLOR 7: LOCATE 0,0
250 END
260 '
270 D=D*(-1)    ← 方向反転
280 DT=DT-50    ← タイマーの値をしだいに小さく(速く)する
290 RETURN
```

行170の **X＝X＋D** が，ボールの進む方向を決めている．すなわち，**D** が＋１のときはボールは右へ，－１のときは左へ進むことになる．この切り変えは，行270で行う．

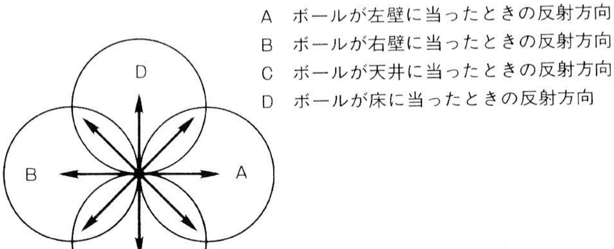
行180，または190で，ボールが壁まできたことを判定し，壁に来たところで行270へ行く．ここで，(＋１)を **D** に掛けているが，これにより，**D** の符号が反転し，ボールが反射する動きをつくることになる．

このプログラムを実行すると，ボールは反射する度ごとにスピードを増すが，これは，行210で，プログラムの次のステップに進むための時間間隙を調整するタイマー（**FOR…NEXT** によるループ文）によって実現している．

3.4 ランダムな反射

ここでは，スクリーン上に表示した枠の中で，ボールが勝手な方向に反射しながら運動するプログラムを体験してみる．

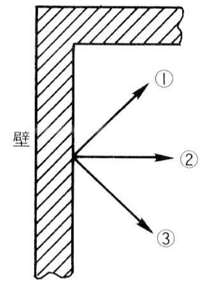
例によって，勝手な方向は，ランダム関数（RND）を用いて，次のように8方向を選択する方法をとることにする．



ボールの反対点におけるボールの進む方向

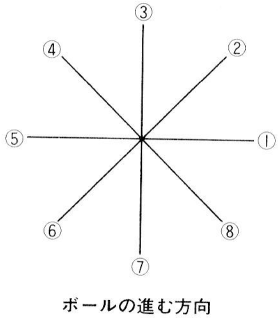
ボールの進む方向は8方向だが，壁での反射を考えると，反射の方向として一度に選択する方向は3通りでよいことがわかる．

Aのケースで考えてみよう．これはボール左壁に当って反射する場合である．ボールは図のように，左向きに進み，左壁に当った場合，もうそれ以上左には行けないことから，ボールの衝突した時点では，図の①〜③の方向を考えればよいことになる．



以下同様で，どの壁に当った場合でも，次の進む方向は3通りを考えればよいことから，ボールの反射による選択は，行340で，1，2，3の3つの数を任意に選ぶ乱数によって実現している．

ボールの反射後の進む方向は，行360以下の8行の8本のサブルーチンで決めている．

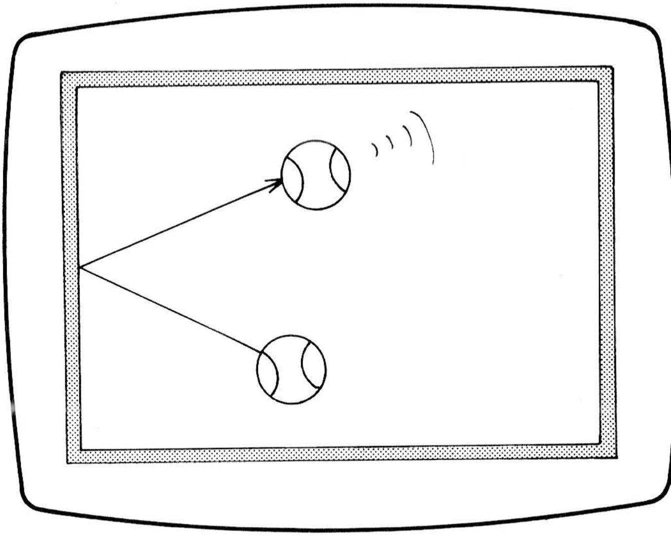


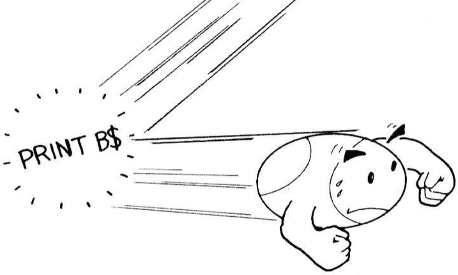
ボールの進む方向

方向	Xの進み幅	Yの進み幅	参照行番号
①	1	0	360
②	1	−1	370
③	0	−1	380
④	−1	−1	390
⑤	−1	0	400
⑥	−1	1	410
⑦	0	1	420
⑧	1	1	430

ボールの進む方向と，X，Yの進み幅の値

```
10 ' P3.4  ﾗﾝﾀﾞﾑﾅ ﾊﾝｼｬ
20 '
30 '
100 SCREEN@ 0 : CLS: DEFINT A-Z
110 LINE(0,4)-(639,195),PSET,4,B  ┐
120 LINE(5,6)-(634,193),PSET,4,B  ├ 枠の表示
130 PAINT(1,5),4                  ┘
140 A$="●":B$=" ":X=40:Y=12:DX=1
150 FOR N=1 TO 1000
160 '
170   X=X+DX : Y=Y+DY
180     IF X<=1 THEN GOSUB 340 ELSE 200
190       ON K GOSUB 360,370,430
200     IF X>=78 THEN GOSUB 340 ELSE 220
210       ON K GOSUB 390,400,410
220     IF Y<=1 THEN GOSUB 340 ELSE 240
230       ON K GOSUB 410,420,430
240     IF Y>=23 THEN GOSUB 340 ELSE 260
250       ON K GOSUB 370,380,390
260 '
270   LOCATE X,Y : PRINT A$  ← ボールの表示
280     FOR T=1 TO 50:NEXT T
290   LOCATE X,Y : PRINT B$ ← 前の位置のボールを消去
300 NEXT N
310 END
320 '
330 'ｻﾌﾞﾙｰﾁﾝ
340 K=INT(RND*3)+1 : RETURN ← 反射する方向の選定
350 '
360 DX=1  :DY=0  : RETURN  ┐
370 DX=1  :DY=-1: RETURN   │
380 DX=0  :DY=-1: RETURN   │
390 DX=-1:DY=-1: RETURN    ├ 反射した場合のX，Yの
400 DX=-1:DY=0  : RETURN   │ 進み幅を決定
410 DX=-1:DY=1  : RETURN   │
420 DX=0  :DY=1  : RETURN  │
430 DX=1  :DY=1  : RETURN  ┘
```




PRINT B$

4 アニメーション



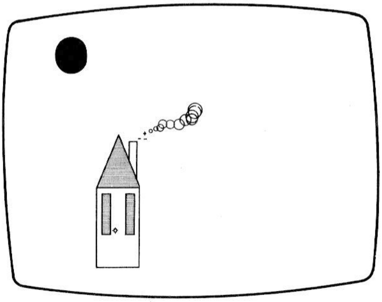
4.1 ケムリを描く

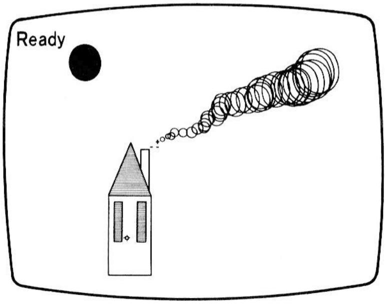
パソコンは楽しいコンピュータである．一般の大型コンピュータやビジネス用のオフィスコンピュータと一寸ちがうのは，簡単に美しいグラフィックスを楽しめるところだろう．

ここで示す例は，その楽しいグラフィックの一例で，家からケムリがモクモク出るところを描いてくれる．

LINE 文で家を描き，CIRCLE 文でケムリを描いている．円を描くとき，その中心位置と半径とを変えると，ケムリに早変り．

```
10 ' P4.1  SMOKING HOUSE
20 '
30 '
100 SCREEN@ 0 :CLS
110 COLOR 7,0
120 CIRCLE(50,25),30,6,,,,F
130 'ｳﾁ ｦ ｴｶﾞｸ
140 LINE(100,185)-(180,125),PSET,7,B  ┐
150 LINE -(140,85),PSET,7             │
160 LINE -(100,125),PSET,7            ├ 家を描く
170 LINE(110,160)-(125,130),PSET,7,B  │
180 LINE(155,160)-(170,130),PSET,7,B  ┘
190 CIRCLE(134,157),5,6                  ← 月を描く
200 LINE(160,105)-(160,90),PSET,7  ┐
210 LINE -(175,90),PSET,7          ├ エントツを描く
220 LINE -(175,120),PSET,7         ┘
230 PAINT(130,120),2,7
240 PAINT(115,150),1,7
250 PAINT(160,150),1,7
260 'ｹﾑﾘ ｦ ｴｶﾞｸ
270 FOR I=1 TO 40
280   X=INT(RND(1)*20)  ┐ ケムリを描くための円の中心位置を指定
290   Y=INT(RND(2)*5)   ┘
300   R=INT(RND(3))+I ← ケムリを描くための円の半径を指定
                                         (しだいに大きく)
310   X1=X1+X : Y1=Y1+Y
320   CIRCLE(167+X1,89-Y1),R,4 ← ケムリを描く
330 NEXT
```




Ready

4.2 ウインキー・カエル

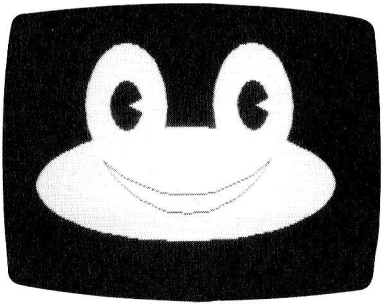
CIRCLE 文を使って，アニメを楽しむことができる．つぎの簡単なプログラムを入力して，実行してみよう．

```
10 '  P4.2  ｶｴﾙ ﾉ ｱﾆﾒ
20 '
30 '
100 SCREEN@ 0 : CLS
110 CIRCLE(320,105),220,4,.15:PAINT(320,105),4,4
120 CIRCLE(420,60),60,4,.6:PAINT(390,60),4,4
130 CIRCLE(230,60),60,4,.6:PAINT(260,60),4,4
140 CIRCLE(320,55),150,0,,.1,.4
150 CIRCLE(230,60),25,0,.6,.05,.95,F
160 '
170 FOR I=1 TO 5
180   CIRCLE(420,60),25,0,.6,.05,.95,F
190   CIRCLE(320,22),200,0,,.135,.365
200    FOR T=1 TO 800:NEXT T
210   CIRCLE(320,22),200,4,,.135,.365
220   PAINT(420,60),4
230   CIRCLE(420,50),40,0,,.1,.4
240    FOR T=1 TO 1500:NEXT T
250   CIRCLE(420,50),40,4,,.1,.4
260 NEXT I
270 GOTO 100
```

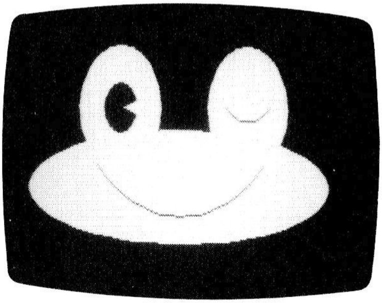
(100～190) 画面Aの表示

(210～230) 画面Bの表示(ウインク)

CIRCLE 文と PAINT 文によって，カエルの顔を描くことができる．この絵を右の写真のように2枚描けるようにして，交互に表示すれば，ウインキー・カエルの出来上り．



画面 A



画面 B

4.3 文字で描く尺取り虫

文字を色々な形で表示することにより，アニメ感覚を実現することもできる．文字を表示するときは，PRINT 文を用いるが，ここで，一寸変わった趣興で，SYMBOL 文を使って，文字を色々な形に変形表示し，面白い効果を出してみよう．まずは，SYMBOL 文の使い方から．

SYMBOL(X, Y), "文字列", 文字の横幅, 文字の縦長,
色コード, 表示角度, { PSET / PRESET / AND / OR / XOR / NOT }

このSYMBOL 文を使って，"COMPUTER" という文字列を，つぎつぎに，延ばしたり，縮めたりしながら，前に進めるプログラムを作ってみた．あたかも，尺取り虫が，進むように，表示されるのは，SYMBOL 文ならではの機能である．

つぎのプログラムを実行すると，しばらくは画面上に何も出てこないが，やがて，左手から，しだいに１文字ずつ姿を現わすので，じっくりとながめてもらいたい．

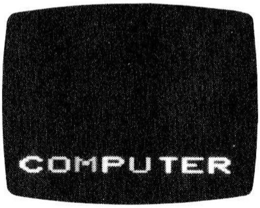
```
10 '  P4.3  "COMPUTER" ｼｬｸﾄﾘﾑｼ ﾉ ｱﾆﾒ
20 '
30 '
100 '
110 SCREENa 0 : CLS
120 DIM H(8),V(8),C$(7),COL(7)
130 FOR I=0 TO 7
140   READ C$(I),COL(I)
150 NEXT I
160  XN=0
170 '
180 FOR N=0 TO 7
190   RESTORE 540
200 '  ｱｸｼｮﾝ 1
210   FOR ST=0 TO 2
220     X=XN+N*112
230     FOR I=0 TO 7
240       READ H(I),V(I)
250     NEXT I
260 '
270     FOR C=0 TO 7
280       DX=8*2^H(C): Y=150-9*2^V(C)
290       SYMBOL(X-DX,Y),C$(C),2^H(C),2^V(C),COL(C)
300       X=X-DX
310     NEXT C
320     FOR T=1 TO 1000 : NEXT T
330     CLS
340   NEXT ST
350 'ｱｸｼｮﾝ 2
360   FOR ST=3 TO 4
370     X=XN-152+N*112
380     FOR I=0 TO 8
390       READ H(I),V(I)
400     NEXT I
410 '
420     FOR C=7 TO 0 STEP -1
430       DX=8*2^H(C+1): Y=150-9*2^V(C)
440       SYMBOL(X+DX,Y),C$(C),2^H(C),2^V(C),COL(C)
450       X=X+DX
460     NEXT C
470     FOR T=1 TO 1000 : NEXT T
480     CLS
490   NEXT ST
500 NEXT N
510 'ﾋｮｳｼﾞｮｳ ﾃﾞｰﾀ
520 DATA R,7,E,6,T,5,U,4,P,3,M,2,O,1,C,7 ← 表示文字データ
530 '
540 DATA 2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0 ┐ のびるモーションの
550 DATA 2,0,2,0,1,1,1,2,1,2,1,1,2,0,2,0 ┤ 横，縦倍率のデータ
560 DATA 2,0,1,1,1,2,0,3,0,3,1,2,1,1,2,0 ┘
570 '
```

```
580 DATA 2,0,2,0,1,1,1,2,1,2,1,1,2,0,2,0,0,0 } ちじむモーションの
590 DATA 2,0,2,0,2,0,2,0,2,0,2,0,2,0,2,0,0,0 } 横，縦倍率のデータ
```

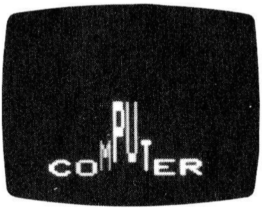
写真の①〜⑤の動きをサイクリックにくり返し実行させるところに注目されたい．

初めはRの文字をピボットに身を縮め，次いでCの文字をピボットにして，ウーンと延びる動作をくり返す．このようにして，尺取り虫のように前進運動をさせるところに工夫をしている．

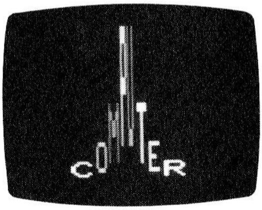
①



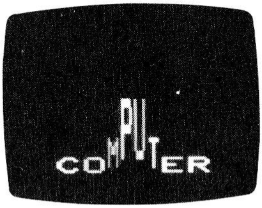
②



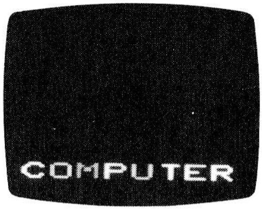
③



④



⑤



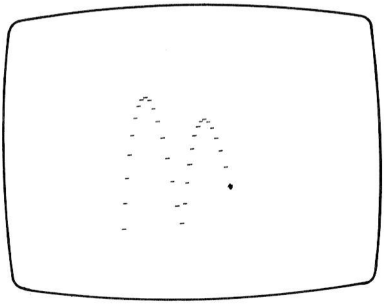
4.4 ボールのゆくえ

楕円の軌跡上に，ボールを描いては消してゆくことにより，ボールの動きを描くことができる．つぎのプログラムは，ボールがはずんで，だんだん，はずみが小さくなり，やがて転がっていく様子を描いたものである．

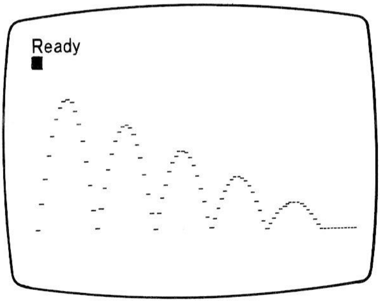
```
10 ' P4.4   ANIMATION OF BOLL MOVEMENT
20 '
30 '
100 SCREEN@ 0 : CLS : LOCATE 25,0
110 X=0
120 FOR A=100 TO 0 STEP -40
130   FOR TH=0 TO 3.14 STEP .2
140     X=X+7
150     Y=150-A*SIN(TH)
160     CIRCLE(X,Y), 5,2
170     PAINT(X,Y),2,2
180     CIRCLE(X,Y),5,2,,,,,PRESET
190   NEXT TH
200 '
210   FOR TH=3.14 TO 6.28 STEP .2
220     X=X+7
230     Y=150-ABS((A-20)*SIN(TH))
240     CIRCLE(X,Y), 5,2
250     PAINT(X,Y),2,2
260     CIRCLE(X,Y),5,2,,,,,PRESET
270   NEXT TH
280 NEXT A
```

（120～180）楕円軌跡の上半分上のボールの動きを表示

（210～260）楕円軌跡の下半分を上・下折り返し反転させ，ボールのはずむ様子を表示



ボールのはずむ様子（途中）



ボールのはずんだ軌跡

5 パターンの作成



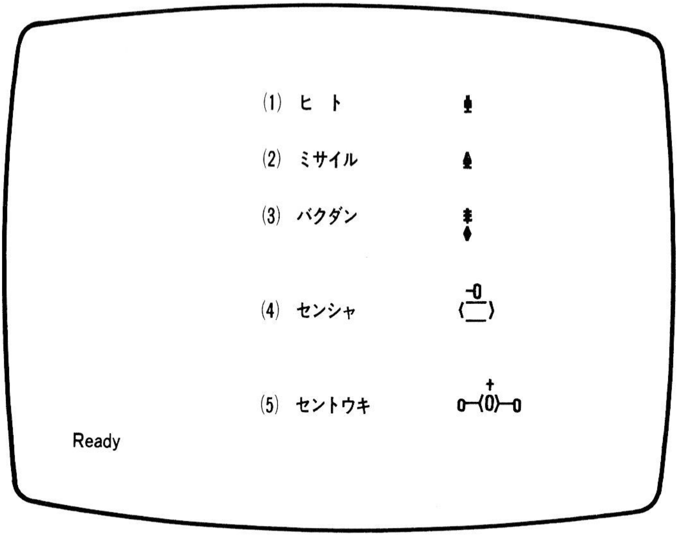
5.1 文字パターンの活用

ゲームには色々なキャラクター・パターンが登場する．精密な図のものやカラーフルなものなど，特徴のあるものが考え出されている．しかし，精密なものだけが良いのではない．次の例は，簡単にパターンを表示する一例を示す．

文字パターンには，ABC……Zからアイウ……ン等の文字や，ハートやスペードなどの文字パターンが使えるようになっている．これらの各種パターンを組み合せることにより，つぎのように簡単なパターンをつくり出すことができる．工夫しだいで，いくらでも面白いパターンが，つぎつぎとスクリーン上に浮びあがってくるはずである．

```
10 '   PS.1ﾓｼﾞﾊﾟﾀｰﾝ ﾉ ｻｸｾｲ
20 '
30 '
100 SCREEN@ 0 : CLS
110 H$="♣"
120 M$="♠"
130 B1$="*":B2$="♦"
140 T1$=" -O":T2$="<￣￣￣>":T3$=" ￣￣￣"
150 J1$="     +":J2$="o--<O>--o"
160 '       ﾋﾄ ﾉ ﾋｮｳｼﾞ
170 LOCATE 50,4 : PRINT H$
180   FOR T=1 TO 500:NEXT T
185 '       ﾐｻｲﾙ ﾉ ﾋｮｳｼﾞ
190 LOCATE 50,7 : PRINT M$
200   FOR T=1 TO 500:NEXT T
205 '       ﾊﾞｸﾀﾞﾝ ﾉ ﾋｮｳｼﾞ
210 LOCATE 50,10 : PRINT B1$
220 LOCATE 50,11 : PRINT B2$
```

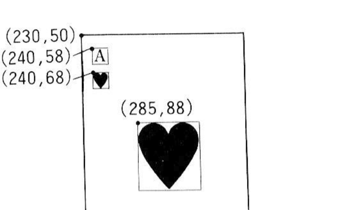
```
230   FOR T=1 TO 500:NEXT T
235 '      ｾﾝｼｬ ﾉ ﾋｮｳｼﾞ
240 LOCATE 50,15 : PRINT T1$
250 LOCATE 50,16 : PRINT T2$
260 LOCATE 50,17 : PRINT T3$
270   FOR T=1 TO 500:NEXT T
275 '      ｾﾝﾄｳｷ ﾉ ﾋｮｳｼﾞ
280 LOCATE 50,20 : PRINT J1$
290 LOCATE 50,21 : PRINT J2$
300   FOR T=1 TO 1000:NEXT T
310 '
320 LOCATE 25,4 :PRINT"(1)  ﾋ ﾄ"
330 LOCATE 25,7 :PRINT"(2)  ﾐｻｲﾙ"
340 LOCATE 25,10:PRINT"(3)  ﾊﾞｸﾀﾞﾝ"
350 LOCATE 25,15:PRINT"(4)  ｾﾝｼｬ"
360 LOCATE 25,20:PRINT"(5)  ｾﾝﾄｳｷ"
```

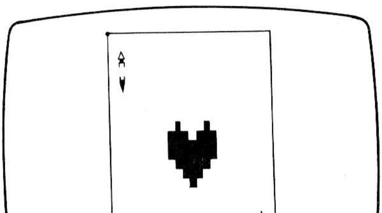


5.2 シンボルの活用

SYMBOL 文を使うと，文字の大きさを色々と変えたり，文字表示をグラフィック座標の任意の場所に表示したり，表示の向きを変えたり，表示そのものにバラエティを持たせることができる．その一例として，トランプのパターンを描いてみよう．

```
10 '   PS.2  ｼﾝﾎﾞﾙ ﾆﾖﾙ ﾄﾗﾝﾌﾟ ﾉ ﾋｮｳｼﾞ
20 '
30 '
100 SCREEN@ 0:COLOR 7,1:CLS
110 LINE(230,50)-(410,150),PSET,7,BF
120 COLOR 2
130 SYMBOL(285,88),"♥",8,4,2,0,PSET
140 SYMBOL(240,58),"A",1,1,2,0,PSET
150 SYMBOL(240,68),"♥",1,1,2,0,PSET
160 SYMBOL(400,142),"A",1,1,2,2,PSET
170 SYMBOL(400,132),"♥",1,1,2,2,PSET
```





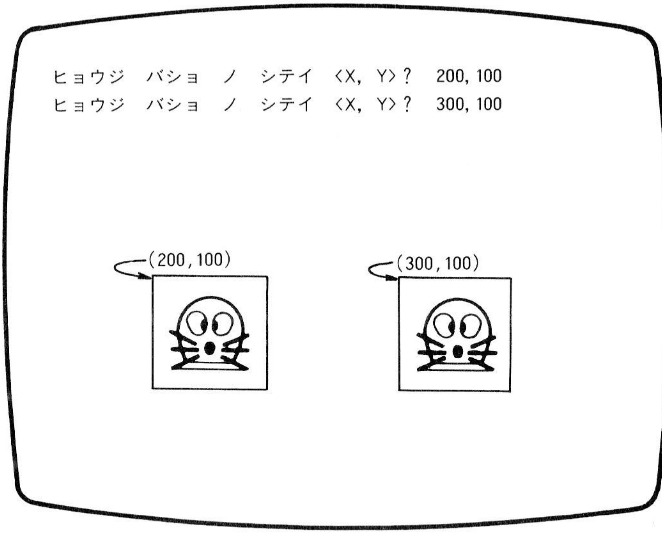
5.3 グラフィック・パターンの保存と表示

より精密なグラフィック・パターンはどのように，表示，保存すればよいのであろうか．ゲームにしても，学習用，ビジネス用でも，より美しい画面構成をするには，より美しいグラフィック図形は欠かせない．

一度作ったグラフィック図形を保存しておき，任意の場所に手軽に表示することができれば素晴しい．このような要望を可能にしてくれたのが，次の GET@，PUT@文である．一度スクリーン上に表示した図形は，GET@文で，前もって準備されている配列変数にとり込むことにより，いつでも PUT@文で，任意の指定場所に表示することができる．

つぎの例は，DATA 文中にとり込んだ，モグラの図柄を，指定場所に表示するプログラムの１例を示している．

```
10 '  PS.3   ﾓｸﾞﾗ ﾉ ﾋｮｳｼﾞ
20 '
30 '
100 SCREEN@ 0 : WIDTH 80: DEFINT A-Z
110 DIM PAT(543)
120 FOR I=0 TO 271
130 READ PAT(I)
140 NEXT I
150 '
160 INPUT"ﾋｮｳｼﾞ ﾊﾞｼｮ ﾉ ｼﾃｲ < X,Y > ";X,Y
170 IF X>579 OR Y>175 THEN PRINT"ﾓｳｽｺｼ
      ﾁｲｻｲ ｱﾀｲ ｦ ﾆｭｳﾘｮｸｼﾃ ｸﾀﾞｻｲ !":GOTO 160
180 '
190 PUT@A(X,Y)-(X+59,Y+23),PAT,PSET
200 GOTO 160
210 '
220 '   ﾓｸﾞﾗ ﾉ ﾃﾞｰﾀ
230 '
240 DATA -1,-1,-1,-1,-1,0,4095,-1,-1024,0,1023,
         -2,0,0
250 DATA 2047,-128,0,0,8191,-4096,0,0,-4,0,0,3,
         -128,0,0,31,-4096,0,0,254,0,3591,0
260 DATA 2016,3,-3844,0,32256,127,4064,7,-8192,
         992,31744,112,0,3075,0,224,0,0,0
270 DATA 32512,0,0,15,-4096,0,0,255,-32768,0,0,
         7936,0,0,0,4064,0,0,127,-256,0,0
280 DATA 4095,-2048,0,1,-256,0,0,0,0,0,0,0,0,0,
         0,0,0,0,0,0,15,-1,0,0,8191,-1,-32768
290 DATA 15,-1,-1,0,1023,-1,-4,0,-128,-16,8176,
         31,-16389,-513,-8320,1015,-33,-16385
300 DATA 31744,32639,-3592,-9,-8185,-2052,3843,
         -130,127,32640,-4065,-2080,2044,-993
310 DATA -31751,-512,32764,1020,1023,-8192,32767,
         -1,-64,56,16383,-1,960,1020,4095
320 DATA -497,-1024,8191,-1,-1,-32768,7,-2,0,7,
         -1,-1,-512,60,2047,-497,-16384,31
330 DATA -1,-512,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
         0,15,-1,0,0,8191,-1,-32768,15,-1
340 DATA -1,0,1023,-1,-4,0,-128,-16,8176,31,
         -16389,-513,-8320,1015,-33,-16385`
350 DATA 31744,32639,-3592,-9,-8185,-2052,3843,
         -130,127,32640,-4065,-2080,2044
360 DATA -993,-31751,-512,32764,1020,1023,-8192,
         32767,-1,-64,56,16383,-1
370 DATA 960,1020,4064,32271,-1024,8191,-1021,
         -1,-32768,7,-16322,0,7,-2,2047
380 DATA -512,60,2047,-497,-16384,31,-1,-512,
         248,8191,-1,-64,-3969,-1,-1,-16,0,0
```



ここで大切なことは，表示図形の大きさ（ドット数）に応じて，表示前に，DIM 文で，必要なだけのメモリを整数型の配列変数として確保しておくということである．


君は アッチ
君はコッチ
GET
PUT

6

これは便利なパターン・ジェネレータ



6.1 パターン・ジェネレータの設計

(1) パターン・ジェネレータの概要

ゲーム・ソフトを作ったり，画面に美しいグラフィック・パターンを使いたいときなど，そのパターン作りには大変な労力を必要とする．ここでは，そんな悩みを解消する，手軽な，パターン作成用のツールを作ってみた．

1文字を1ドットとして表示し，画面全体を使って，79×23ドットのカラーパターンを作ることにする．

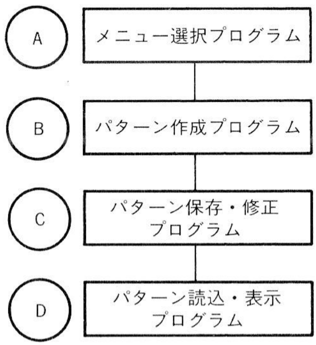
始めに，パターンのサイズを決め，"＋"のマークで表示するカーソルの位置に"●"を使って，ドットをカラーで拡大表示し，パターンを絵を描く感覚で作っていく．完成したパターンは，フロッピー・ディスクに保存することができ，いつでも，必要なときにとり出して，表示できるようにしておく．

本文中で使う，いくつかのパターンは，このパターン・ジェネレータを使って作り出したもので，仲々便利なものである．

(2)　プログラムの構成

本プログラムは，全体的に４つの部分に分かれている．以後に，各部分について，細部の説明をしておこう．

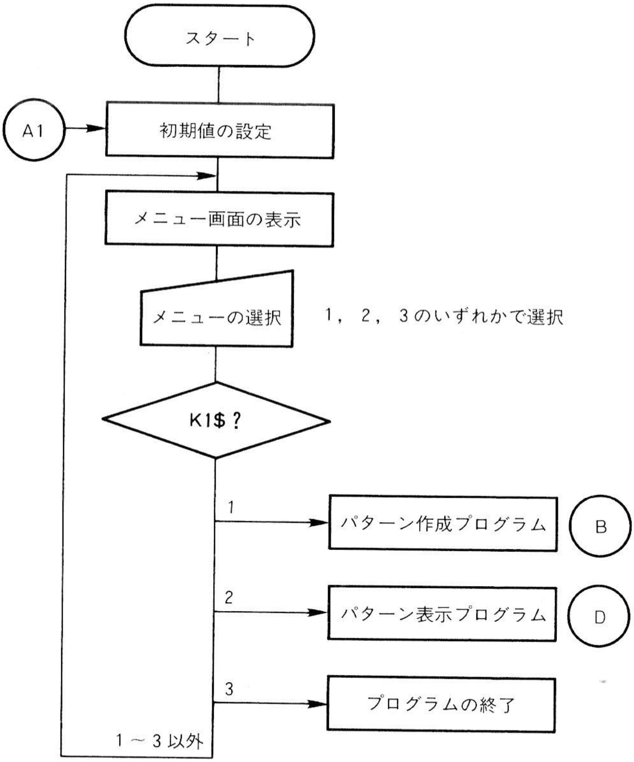
プログラムの全体構成



Ⓐ　メニュー選択プログラム

この部分は，全体のプログラムの導入部で，以後のいろいろなプログラムに分岐させる役割を果たしている．

Ⓐ メニュー選択プログラムの流れ



〈変数の説明〉

SW1：配列変数の二重定義防止用のスイッチ．配列変数 CDM，PDM 1 の定義がされていれば SW 1＝1，未定義であれば SW 1＝0

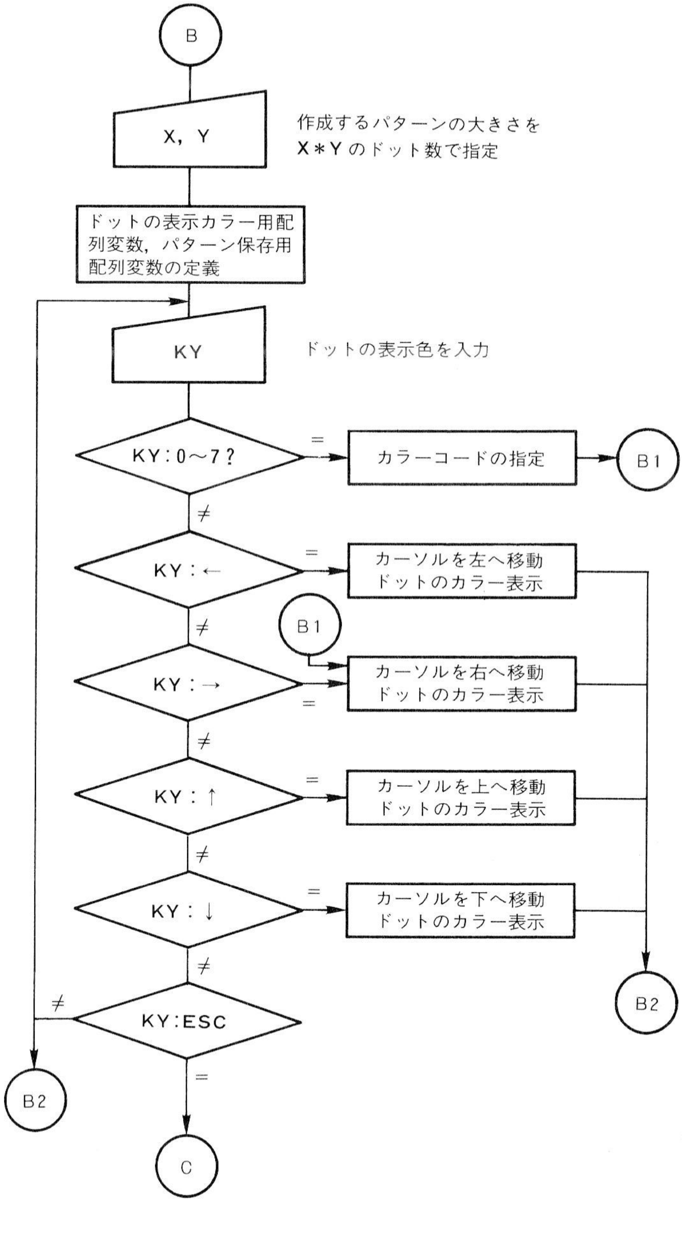
```
10 '  P6.1    Pattern Generater
20 '
30 '
1000 DEFINT A-Z:SCREEN@ 0
1010 '
1020 '      ﾒﾆｭｳ ｶﾞﾒﾝ
1030 '
1040 SW1=0
1050 '
1060 WIDTH 40 : COLOR 6
1070 PRINT"*****    ﾊﾟﾀｰﾝ ｻｸｾｲ ﾌﾟﾛｸﾞﾗﾑ    ****":PRINT
1080 PRINT"ﾂｷﾞﾉ ﾄﾞﾚｶ ｦ ｴﾗﾝﾃﾞ ｸﾀﾞｻｲ":PRINT
1090 PRINT"1. ······  ﾊﾟﾀｰﾝ ﾉ ｻｸｾｲ"
1100 PRINT"2. ······  ﾊﾟﾀｰﾝ ﾉ ﾋｮｳｼﾞ"
1110 PRINT"3. ······  END":PRINT
1120 K1$=INPUT$(1)
1130 ON VAL(K1$) GOTO 1180,2120,2450
1140 IF VAL(K1$)=<0 OR VAL(K1$)=>4 THEN 1060
```

Ⓑ パターン作成プログラム

この部分は，パターン・ジェネレータの主要な機能を果たしている．XとYで指定した範囲の広さのなかで，カーソルを自由に動かし，数字キーでカラー・コードを指定し，美しいグラフィック・パターンを作成する部分を実現するプログラムである．

カーソルマークの"＋"を自由に移動させたり，前の表示結果を手軽に変更できるようにするところに，やや工夫を要した．

Ⓑ パターン作成プログラム



〈変数の説明〉

PS1　：**X＊Y** ドットのパターンを格納する配列のサイズ
CDM　：パターン内の各ドットのカラーコードを格納する配列
PDM　：パターン情報を格納（**GET@**）するための配列変数
CSX　：ドットを表示する場所を示すカーソルの **X** 座標位置
CSY　：ドットを表示する場所を示すカーソルの **Y** 座標位置
KY　：ドットのカラーコード **OR** カーソル移動キー入力用変数
FCSX：**CSX** の１つ前の **X** 座標位置を示す
FCSY：**CSY** の１つ前の **Y** 座標位置を示す
P$　：ドットの位置を示すカーソルマーク

```
1150 '
1160 '      ﾊﾟﾀｰﾝ ﾉ ｻｸｾｲ
1170 '
1180 CLS : LOCATE 15,4
1190 PRINT"ﾊﾟﾀｰﾝ ﾉ ｻｸｾｲ"
1200 LOCATE 5,8:INPUT "< ﾊﾟﾀｰﾝｻｲｽﾞ? ; X,Y>";X,Y
1210  IF X>80 OR Y>24 THEN 1200
1220  PS1=((X*Y+7)/8)*3
1230 IF SW1=1 THEN ERASE CDM,PDM1
1240  DIM CDM(X-1,Y-1),PDM1(PS1) : SW1=1
1250 '
1260 WIDTH 80 : COLOR 7
1270 LINE(0,0)-(8*X,8*Y),PSET,7,B
1280 CSX=0:CSY=0:LOCATE CSX,CSY:PRINT"+";:LOCATE CSX,CSY
1290 '
1300 KY=ASC(INPUT$(1))
1310  IF KY>=&H30 AND KY<=&H37 THEN P$="+" : GOTO 1740
1320  IF KY=&H1D THEN 1370
1330  IF KY=&H1C THEN 1470
1340  IF KY=&H1E THEN 1570
1350  IF KY=&H1F THEN 1650
1360  IF KY=&H1B THEN 1800 ELSE 1300
1370 ' <--
1380 CSX=CSX-1 : FCSX=CSX+1 : FCSY=CSY
1390 IF CSX=-1 THEN CSX=X-1 : FCSX=0 ELSE 1430
1400 COL=CDM(FCSX,FCSY):COLOR COL
1410 LOCATE FCSX,FCSY:PRINT "●";
1420 GOTO 1570
1430 LOCATE FCSX,FCSY:COLOR CDM(FCSX,FCSY)
1440 PRINT "●"; : LOCATE CSX,CSY : COLOR 7 : PRINT"+";
1450  LOCATE CSX,CSY
1460 GOTO 1300
1470 ' -->
1480 CSX=CSX+1 : FCSX=CSX-1 : FCSY=CSY
1490 IF CSX=X THEN CSX=0 : FCSX=X-1 ELSE 1530
1500 COL=CDM(FCSX,FCSY):COLOR COL
1510 LOCATE FCSX,FCSY:PRINT "●";
```

```
1520 GOTO 1650
1530 LOCATE FCSX,FCSY:COLOR CDM(FCSX,FCSY)
1540 PRINT "●"; : COLOR 7 : PRINT "+";
1550  LOCATE CSX,CSY
1560 GOTO 1300
1570 ' UP
1580 CSY=CSY-1 : FCSX=CSX : FCSY=CSY+1
1590 IF CSY=-1 THEN CSY=Y-1 : FCSY=0
1600 COL=CDM(FCSX,FCSY):COLOR COL
1610 LOCATE FCSX,FCSY:PRINT "●";
1620 COLOR 7:LOCATE CSX,CSY:PRINT P$;:P$="+"
1630  LOCATE CSX,CSY
1640 GOTO 1300
1650 ' DOWN
1660 CSY=CSY+1 : FCSX=CSX : FCSY=CSY-1
1670 IF CSY=Y THEN CSY=0 : FCSY=Y-1
1680 COL=CDM(FCSX,FCSY):COLOR COL
1690 LOCATE FCSX,FCSY:PRINT "●";
1700 COLOR 7:LOCATE CSX,CSY:PRINT P$;:P$="+"
1710  LOCATE CSX,CSY
1720 GOTO 1300
1730 '      ｶﾗｰｺｰﾄﾞ ﾉ ｼﾃｲ
1740 COL=KY-&H30
1750 COLOR COL
1760  PRINT "●";:COLOR 7: PRINT P$; : LOCATE CSX+1,CSY
1770 CDM(CSX,CSY)=COL
1780 GOTO 1470
```

Ⓒ　パターンの保存・修正プログラム

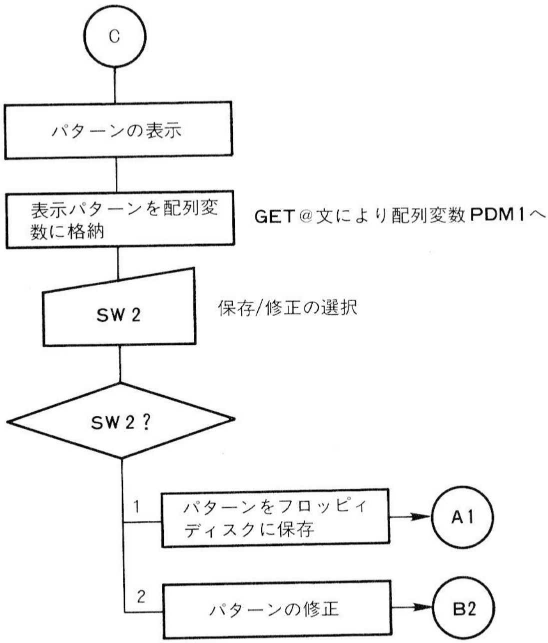
パターンができ上ると，次に大切なことは，正確にパターンの情報をファイルの形式で，フロッピィ・ディスクに保存できるようにしておくことである．そのためには，でき上ったパターンを，一旦，パターン格納用の配列変数に GET@文でとり込んでおき，その配列変数の内容をデータとしてフロッピィにファイル名を指定して出力するという方法をとる．

このようにしておくと，必要なときには，ファイルを呼び出して，配列変数にとり込み，PUT@文を使って，パターンを画面上の任意の場所に表示をするということで，いつでも再現させることができるのである．

一方，画面上でのパターンは，いつでも手軽に修正ができるようにしておいた方が良い．これは一応実現させておいたが本プログラムではフロッピィに保存したパターンについては，修正できるようになっていない．

これはそう難かしいことではない．フロッピィに出力するときに，いっしょにカーラー情報用の配列変数 CDM も保存しておくように書き直しておけば，いつでも修正ができるようになる．これは読者への課題として残しておこう．

Ⓒ パターンの保存・修正プログラム



```
1790 '      ﾊﾟﾀｰﾝ ﾉ ｻｸｾｲ
1800 CLS
1810 FOR I=0 TO X-1
1820  FOR J=0 TO Y-1
1830   PSET(I,J,CDM(I,J))
1840 NEXT J,I
1850 '      ﾊﾟﾀｰﾝ ﾉ ｷﾛｸ
1860 LOCATE 50,0:PRINT"ﾊｲﾚﾂ ﾉ ｻｲｽﾞ = ";PS1
1870 GET@A (0,0)-(X-1,Y-1),PDM1,G
1880 PUT@A(300,100)-(299+X,99+Y),PDM1,PSET
1890 COLOR 7
1900 LOCATE 0,20
1910 INPUT"<1: ﾎｿﾞﾝ / 2: ｼｭｳｾｲ>";SW2
1920 ON SW2 GOTO 1930,2030
1930 '    ﾊﾟﾀｰﾝ ﾉ ﾎｿﾞﾝ
1940 WIDTH 40 : COLOR 4
1950 LOCATE 15,0:PRINT"ﾊｲﾚﾂ ﾉ ｻｲｽﾞ = ";PS1
1960 LOCATE 5,10:INPUT"<ﾊﾟﾀｰﾝ ﾉ ﾅﾏｴ?:8ﾓｼﾞ ｲﾅｲ>";PNA1$
1970 OPEN "O",#1,PNA1$
1980 FOR I=0 TO PS1
1990  PRINT #1,PDM1(I)
2000 NEXT
2010 CLOSE
2020 GOTO 1060
2030 '
2040 WIDTH 80
2050 FOR DY=0 TO Y-1
2060   FOR DX=0 TO X-1
2070     LOCATE DX,DY:COLOR CDM(DX,DY):PRINT "●";
2080 NEXT DX,DY
2090 GOTO 1270
```

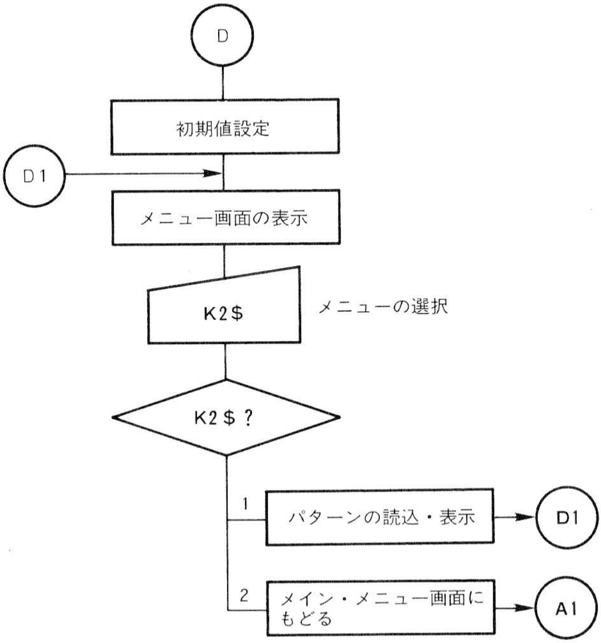
Ⓓパターン読み込み・表示プログラム

フロッピィに保存したパターンを再現させるプログラムである．

パターン情報をデータ・ファイルとしてフロッピィ上に保存してあるので，その内容を再び，パターン表示用の配列変数にとり込めば，そのまま再表示ができるわけだ．

ここで，必要なことは，前に保存したパターンの大きさヨコのドット数（X）と縦のドット数（Y），および，保存したときのファイルの名称を正しく覚えておかなければならないということである．

Ⓓ パターン読込・表示プログラム



〈変数の説明〉
SW3　：パターン読込み用配列変定義/未定義用スイッチ
PDM2：パターン読込み用配列変数

```
2100 '
2110 '    ﾊﾟﾀｰﾝ ﾉ ﾖﾐｺﾐ･   ﾋｮｳｼﾞ
2120 '
2130 SW3=0
2140 DEFINT A-Z
2150 '
2160 WIDTH 40 : COLOR 6: LOCATE 5,10
2170 PRINT"*****    ﾊﾟﾀｰﾝ ﾉ ﾋｮｳｼﾞ    ****":PRINT
2180 LOCATE 5,12:PRINT"<1: ﾊﾟﾀｰﾝ ｦ ﾋｮｳｼﾞ ｽﾙ>
2190 LOCATE 5,13:PRINT"<2: ﾒﾆｭｳ ｶﾞﾒﾝ ﾆ ﾓﾄﾞﾙ>
2200 K2$=INPUT$(1)
2210 IF VAL(K2$)=<0 OR VAL(K2$)>=3 THEN 2160
2220 ON VAL(K2$) GOTO 2240,1060
```

```
2230 '
2240 LOCATE 5,15:INPUT "< ﾊﾟﾀｰﾝｻｲｽﾞ? ; X,Y>";X,Y
2250  IF X>80 OR Y>24 THEN 2240
2260  PS2=((X*Y+7)/8)*3
2270 IF SW3=1 THEN ERASE PDM2
2280  DIM PDM2(PS2) : SW3=1
2290 COLOR 4:PRINT
2300 LOCATE 5,17:INPUT"<ﾊﾟﾀｰﾝ ﾉ ﾅﾏｴ?:8ﾓｼﾞ ｲﾅｲ>";PNA2$
2310 '
2320 OPEN "I",#1,PNA2$
2330   FOR I=0 TO PS2
2340   INPUT#1,PDM2(I)
2350   NEXT I
2360 WIDTH 80 : COLOR 7
2370 LOCATE 50,0 :PRINT"ﾊﾟﾀｰﾝ ﾉ ﾅﾏｴ = ";PNA2$
2380 LOCATE 50,1:PRINT"ﾊｲﾚﾂ ﾉ ｻｲｽﾞ = ";PS2
2390 PUT@A(400,100)-(399+X,99+Y),PDM2,PSET
2400 LOCATE 0,24
2410 PRINT"ﾂｷﾞﾆｽｽﾑﾄｷ ﾊ , <ｷｰ> ｦ ﾄﾞﾚｶ ｵｼﾃ ｸﾀﾞｻｲ !"
2420 PK$=INPUT$(1)
2430 CLOSE : GOTO 2160
2440 '
2450 WIDTH 80 : COLOR 7 : END
```

以上で，パターン・ジェネレータのプログラムの説明を終るが，ここで追加的に，もう１つプログラムを紹介しておきたい．

作成したパターンを別のプログラム中で使いたい場合，パターンのデータを，DATA 文で，プログラム中に書き込んでおけば便利である．そんな目的のために，一度フロッピィ・ディスクの中にファイルの形で保存されたパターン・データを，DATA 文の形にプリンターに出力するプログラムを作ってみた．つぎのプログラムがそれである．簡単なので，知っておいて損はないだろう．

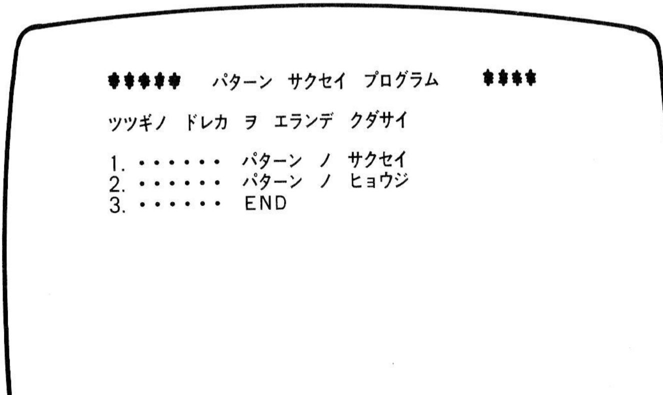
```
10 '
20 '   ﾊﾟﾀｰﾝ ﾉ ﾖﾐｺﾐ･  ﾋｮｳｼﾞ
30 '
100 DEFINT A-Z: CLS
110 '
120 LOCATE 5,8:INPUT "< ﾊﾟﾀｰﾝｻｲｽﾞ? ; X,Y>";X,Y
130  IF X>80 OR Y>24 THEN 120
140  PS2=((X*Y+7)/8)*3
150  DIM PDM2(PS2)
160 COLOR 4:PRINT
170 LOCATE 5,17:INPUT"<ﾊﾟﾀｰﾝ ﾉ ﾅﾏｴ?:8ﾓｼﾞ ｲﾅｲ>";PNA2$
180 '
190 OPEN "I",#1,PNA2$
200   FOR I=0 TO PS2
210   INPUT#1,PDM2(I)
220   NEXT I
230 WIDTH 80 : COLOR 7
240 LOCATE 50,0 :PRINT"ﾊﾟﾀｰﾝ ﾉ ﾅﾏｴ = ";PNA2$
250 LOCATE 50,1:PRINT"ﾊｲﾚﾂ ﾉ ｻｲｽﾞ = ";PS2
260 PUT@A(400,100)-(399+X,99+Y),PDM2,PSET
270 LOCATE 0,24
280 PRINT"ﾂｷﾞﾆｽｽﾑﾄｷ ﾊ , <ｷｰ> ｦ ﾄﾞﾚｶ ｵｼﾃ ｸﾀﾞｻｲ !"
290 PK$=INPUT$(1)
300 CLOSE
310 '
312 N=1000
320 FOR I=0 TO PS2 STEP 8
330   N=N+10
340   LPRINT:LPRINT STR$(N);" ";"DATA ";
350     FOR J=0 TO 7
360       LPRINT PDM2(I+J);
365       IF J<>7 THEN LPRINT ",";
370     NEXT J
380 NEXT I
```

（340〜380行）行番号1000からDATA文として，パターン用のデータをプリンターに打ち出す

6.2 パターン・ジェネレータを使ってみる

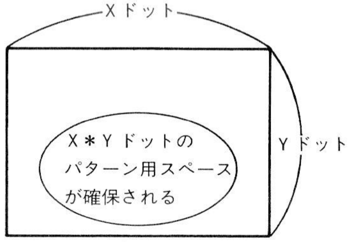
6．1で示したプログラムを，一寸大変だが正確に入力する．入力したら，すぐに，安全のためプログラムをセーブしておく．ここまでが準備の段階で，全てパーフェクトに完了したら，実行してみよう．

まず，メニュー画面が出てくる．



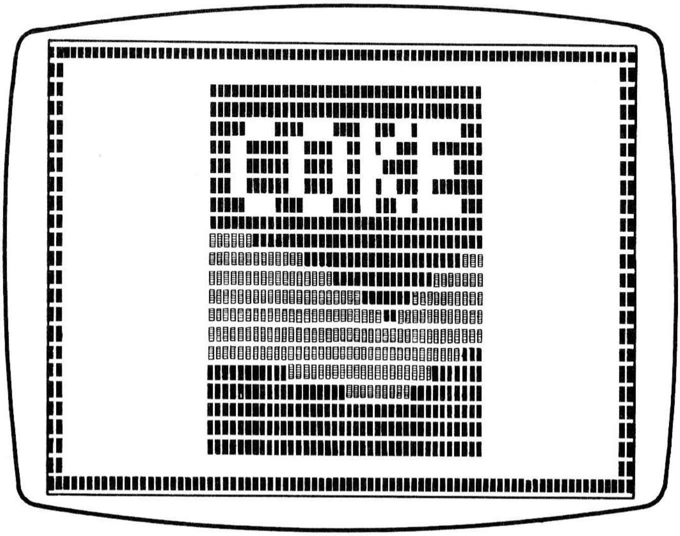
メニュー

パターンを新たに作るので，迷わず，1を選択する．すると，パターンのサイズを聞いてくる．



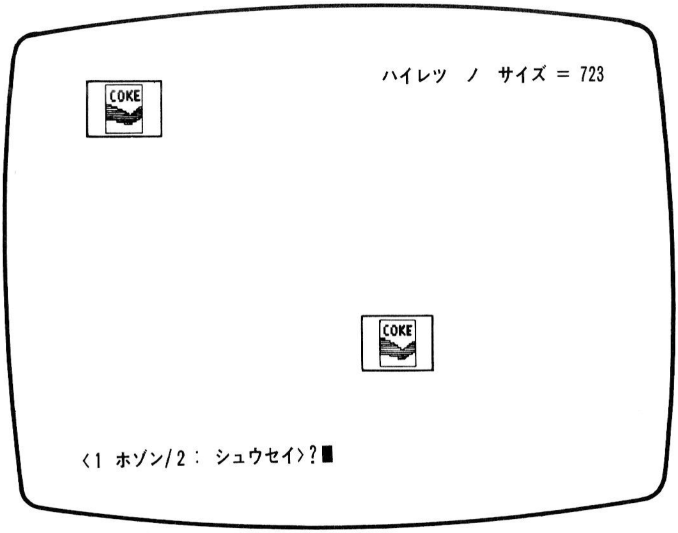
ここまで終れば，すぐにでもパターン作成にとりかかれる．カーソルキーを使い，カーソル表示用の "＋" マークを希望する場所に持っていく．そこで，数字キーを0～7の範囲で押すと，対応するカラーコードで，"●"印が表示される．これをくり返し行うことによって，タイル・パターンができ上る．

つぎにその一例を示す．



完成画面

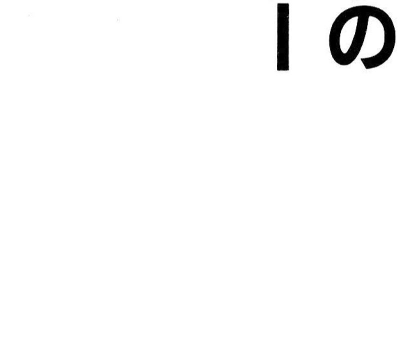
これで，パターンの作成は完了したことになる．次いで，<ESC>キーを押すと，画面上のパターンがパターン格納用配列変数に保存され，完成したパターンを，ドットで表示してくれる．前のタイルパターンが，つぎのように，ドット・パターンとなるのである．これがパターンの完成図である．



この段階では，いつでも，もとのタイルパターンをよび出して，パターンの修正をすることができる．キャンバスに向っているように工夫して，自由に美しいパターンを描き出してみよう．

これでよいとなれば，一旦メニュー画面にもどり，フロッピィに保存して，一連の処理が完了ということになる．

7
面白表示のハウ・ツー



7.1 同一場所に書いては消す

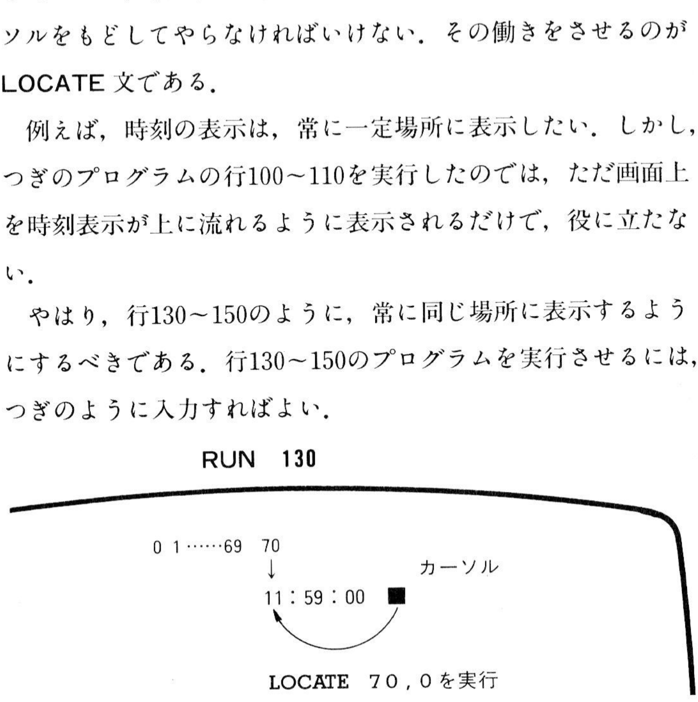
時刻，経過時間，得点，メッセージなどは，全てスクリーン上の特定場所に，いつでも同じように表示することが多い．こんな目的のためには，LOCATE 文を使うことによって効果的に表示場所をコントロールすることができる．

画面上に文字を表示すると，カーソルが移動するので，つぎに表示するときは，前の表示場所の次の場所から表示される．したがって，常に同じ場所に表示するためには，その都度カーソルをもどしてやらなければいけない．その働きをさせるのがLOCATE 文である．

例えば，時刻の表示は，常に一定場所に表示したい．しかし，つぎのプログラムの行100～110を実行したのでは，ただ画面上を時刻表示が上に流れるように表示されるだけで，役に立たない．

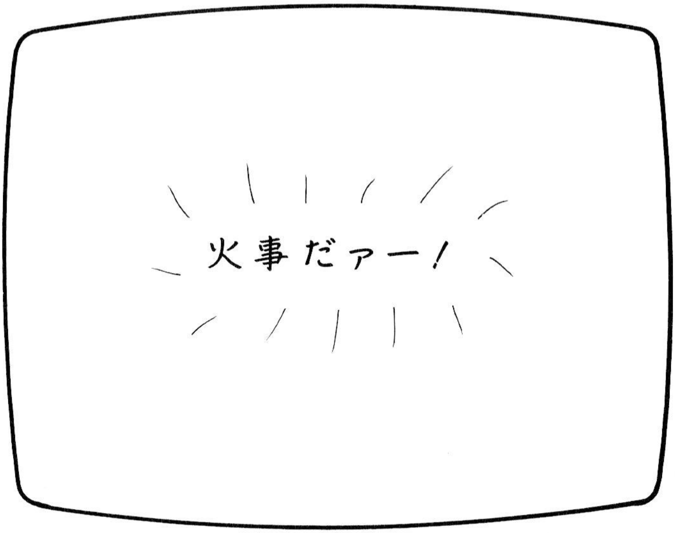
やはり，行130～150のように，常に同じ場所に表示するようにするべきである．行130～150のプログラムを実行させるには，つぎのように入力すればよい．

RUN　130



このプログラムは，それぞれ行100，130，170，250からはじまる4つのプログラムからできている．行170～230は，同じ場所に表示するなかでも，やや表示方法を工夫して，点滅させるようにしたものである．表示を特に注意させたいときなどに用いられる方法である．

プログラム上で，同じ場所に書いたり消したりすればよいわけで，消すためには，書いた場所に，同じ文字数のスペースを表示するという方法をとる．これをくり返せば点滅することになる．点滅の早さは，行190，200に入れたことによって，簡単に調節することができる．



さて，最後の行250以後のプログラムだが，これは，**SYMBOL** 文を使って，**LOCATE** 文より，もっと細かく，表示位置をコントロールして表示する例を示す．このプログラムを実行すると，FM77AV の字がつぎつぎと画面から突び出してくるように，だんだん大きくなりながら，ズーム・アップしてくる様子が見られる．

```
10 '   P 7.1
20 '
30 '
100 PRINT TIME$
110 GOTO 100
120 '
130 LOCATE 70,0
140 PRINT TIME$
150 GOTO 130
160 '
170 LOCATE 35,12
180 PRINT "TIME IS UP !!"
190 FOR T=1 TO 50:NEXT T
200 LOCATE 35,12
210 PRINT "             "
220 FOR T=1 TO 50:NEXT T
230 GOTO 170
240 '
250 CLS
260 FOR N=1 TO 6
270   READ A$
280   COLOR N
290   FOR I=1 TO 25
300     X=320-I*4:Y=100-I*3
310     SYMBOL(X,Y),A$,I,I,N
320       FOR T=1 TO 50:NEXT T
330     CLS
340   NEXT I
350 NEXT N
360 DATA F,M,7,7,A,V
```

7.2 文字パターンを使って表示する

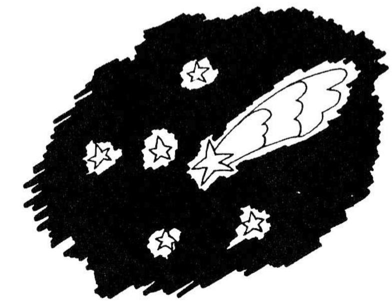
文字をカラフルに表示するだけで，なかなか面白い表示効果を出すことができる．ここでは，乱数発生用の RND 関数と，表示場所指定の LOCATE 文と，色指定用の COLOR 文を使って，2通りのパターン表示をしてみよう．

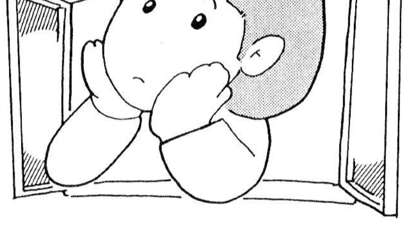
初めは，ごく簡単な例で，夜空の星に似せて，画面上のでたらめな位置に，星をいろいろな色で表示しては消すプログラムを実行してみる．行120～180までがそれである．

ロマンチックな星空とまではいかないが，RND 関数を使うと，このようにランダムな位置にパターンを表示させることができることが理解されると思う．

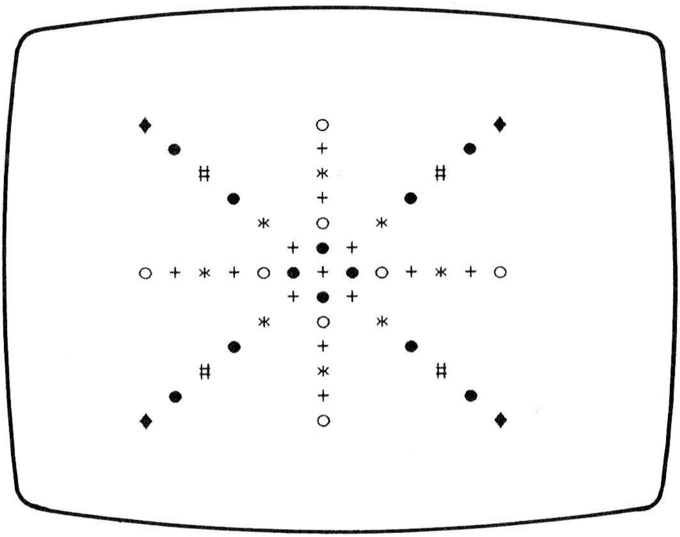
```
10 '  P7.2
20 '
30 '
100 ' ﾖｿﾞﾗ ﾉ ﾎｼ
110 '
120 WIDTH 80 : CLS
130 FOR N=1 TO 50
140   LOCATE 80*RND,23*RND:COLOR 7*RND+1:PRINT"*";
150    FOR I=1 TO 40
160      LOCATE 80*RND,23*RND:PRINT" ";
170    NEXT I
180 NEXT N
190 '
200 LOCATE 24,10:PRINT"ﾂｷﾞﾆｽｽﾑ ---> ｽｷﾅ <ｷｰ> ｦ ｵｽ｡"
210 K$=INPUT$(1)
220 '
```

```
230 '  ﾊﾞｸﾊﾂ
240 CLS:X=40:Y=12
250 FOR N=1 TO 6
260   READ A$,B$,C$
270 '
280   LOCATE X,Y:COLOR INT(7*RND)+1:PRINT A$;
290   LOCATE X+2*N,Y-N:COLOR INT(7*RND)+1:PRINT B$;
300   LOCATE X-2*N,Y-N:COLOR INT(7*RND)+1:PRINT B$;
310   LOCATE X-2*N,Y+N:COLOR INT(7*RND)+1:PRINT B$;
320   LOCATE X+2*N,Y+N:COLOR INT(7*RND)+1:PRINT B$;
330 '
340   LOCATE X+2*N,Y  :COLOR INT(7*RND)+1:PRINT C$;
350   LOCATE X,Y-N    :COLOR INT(7*RND)+1:PRINT C$;
360   LOCATE X-2*N,Y  :COLOR INT(7*RND)+1:PRINT C$;
370   LOCATE X,Y+N    :COLOR INT(7*RND)+1:PRINT C$;
380 NEXT N
390 COLOR 7,0
400 '
410 DATA *,+,●,+,*,○,*,●,+,+,#,*,*,●,+,+,◆,○
```





行240～410は，同じように文字パターンをカラフルに表示する例だが，表示方法を中心から外へ順次表示するように工夫している．このサンプルをもとにすると，花火や，ミサイルの命中シーン等に応用することができる．



7.3 文字パターンの移動

文字パターンを組み合せて表示すれば，それだけで一つの面白いパターンを作り出すことができることは前で学んだ(5．1参照)．このようにして作ったパターンを１組のセットで移動させることができれば，丁度文字で描いた絵を動かすことになる．

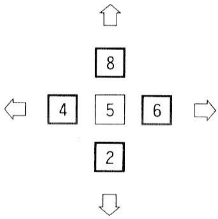
つぎのプログラムで，その１例を試してみよう．この例は，二行を使って描いたジェット機の文字絵である．この文字絵を１組にして，数字キーを押すことで簡単に移動させることができる様子を確認してもらいたい．移動のためのキー操作は，つぎの絵のようになっている．

たった10行のプログラムで，ジェット機を動かすことができるのだ．これをゲームプログラムに組み込めば，自作ゲームの第１作ができ上ることになる．

```
10 '   P7.3  ﾓｼﾞﾊﾟﾀｰﾝ ﾉ ｲﾄﾞｳ
20 '
30 '
100 WIDTH 80
110 J1$="     +":J2$="o---<O>---o"
120 X=40:Y=12
130 LOCATE X,Y  :PRINT J1$;
140 LOCATE X,Y+1:PRINT J2$;
150 K$=INKEY$
160 IF K$="6" THEN X=X+3 : GOTO 200
170 IF K$="4" THEN X=X-3 : GOTO 200
180 IF K$="8" THEN Y=Y-1 : GOTO 200
190 IF K$="2" THEN Y=Y+1 : GOTO 200 ELSE 150
200 CLS : GOTO 130
```

キー操作と

パターンの移動方向



7.4 ジェット機の操縦

キー操作によるジェット機の移動ができるようになると，もう少し本格的な操縦をやってみたくなる．そこで，ここでは，一寸とぜいたくをして，前に出てきたパターン・ジェネレータ（6．1参照）を使って，もう少し精密なジェット機のパターンを作り，いかにも本格的なジョイスティックを使って，操縦用の練習用プログラム（シミュレータ）を作ってみよう．

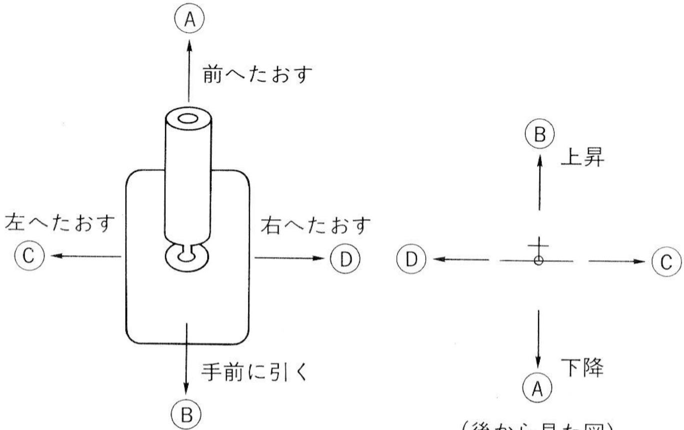


ジョイスティック

まず，このシミュレータで行う操縦のルールを説明しておこう．

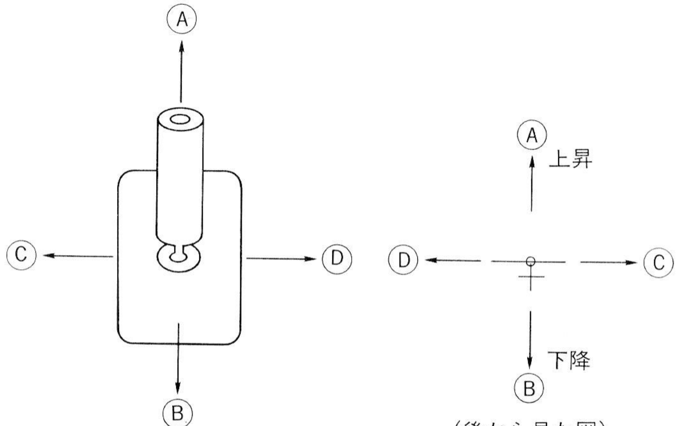
操縦桿の働きをするジョイスティックとジェット機の動力の関係は，つぎの図で示すようにしてある．実際にはこんな単純にはいかないが，このプログラムで方向感覚だけでは模疑的に楽しめるようにしてある．

① 正常飛行時



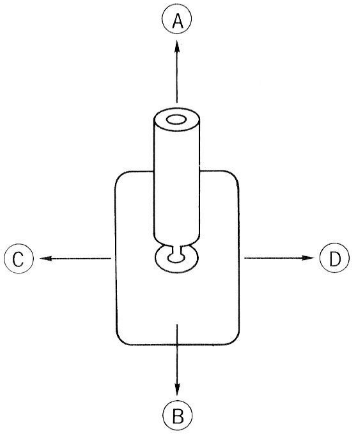
（後から見た図）

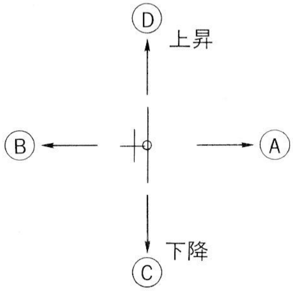
② 背面飛行時



（後から見た図）

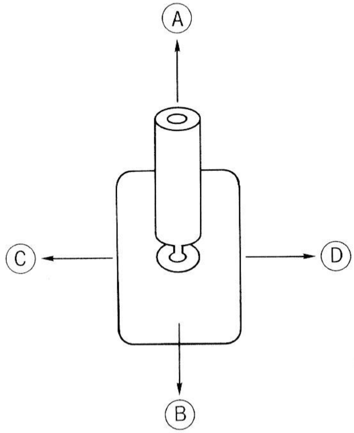
③ 左90°回転飛行

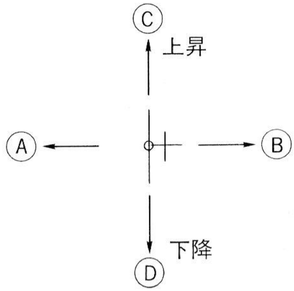




(後から見た図)

④ 右90°回転飛行

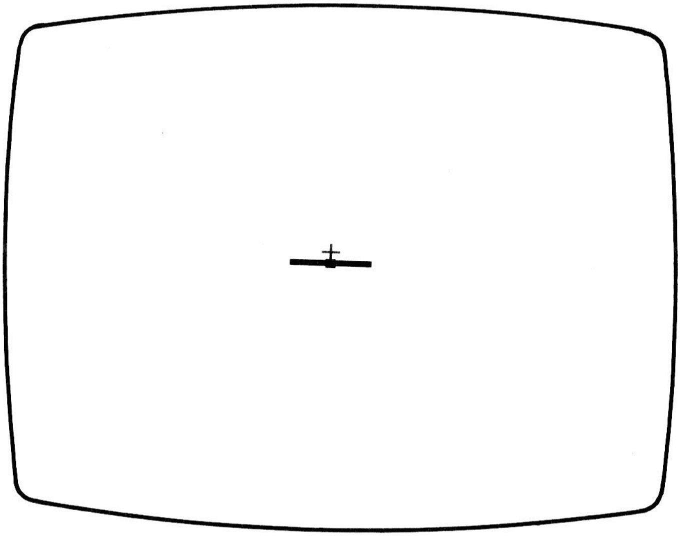




(後から見た図)

つぎにこのプログラムの使い方を示しておこう．

プログラムを実行すると，つぎのような画面が出てくる．



このままでは何も実行しない．プログラムをスタートするには，ジョイスティックの上にある押しボタンを押すと，ジェット機は飛行運動を開始する．

ジョイスティックの上のボタンを押すと，ジェットは90°ずつ左に回転し，ジョイスティックの下についているもう１つのボタンを押すと，同じく90°ずつ右に回転するので，姿勢を色々と変えることによって，姿勢に応じた飛行操縦感覚を体験することができる．

```
10 '  P7.4  FRIGHT SIMULATER
20 '
30 '
1000 SCREEN@ 0 : COLOR 7,1 : CLS
1010 DEFINT A-Z
1020 X=320:Y=100:FC=-1:N=-1
1030 DIM JET1(123),JET2(115),JET3(123),
       JET4(115)
1040 '
1050 '  ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ
1060 FOR I=0 TO 63
1070   READ JET1(I)
1080 NEXT I
1090 FOR I=0 TO 63
1100   READ JET2(I)
1110 NEXT I
1120 FOR I=0 TO 63
1130   READ JET3(I)
1140 NEXT I
1150 FOR I=0 TO 55
1160   READ JET4(I)
1170 NEXT I
1180 '
1190 PUT@A(X,Y)-(X+39,Y+7),JET1,PSET
1200 '
1210 '  ﾌﾗｲﾄ ｺﾝﾄﾛｰﾙ
1220 '
1230 ST1=STRIG(1):ST2=STRIG(3)
1240 '
1250 IF ST1<>-1 THEN 1300
1260   IF N=4 THEN N=0
1270     N=N+1
1280     FC=N MOD 4
1290 '
1300 ON FC+1 GOSUB 1430,1530,1630,1730
1310 '
1320  IF ST2<>-1 THEN 1370
1330    IF N=0 THEN N=4
1340      N=N-1
1350      FC=N MOD 4
1360 '
1370  ON FC+1 GOSUB 1430,1530,1630,1730
1380 '
1390 GOTO 1230
1400 '
1410 '  ﾌﾗｲﾄ ｺﾝﾄﾛｰﾙ ｻﾌﾞﾙｰﾁﾝ
1420 '
1430 '  ｽｲﾍｲ ﾋｺｳ
1440 D=STICK(1)
1450 LINE(X,Y)-(X+39,Y+14),PSET,1,BF
1460   IF D=1 THEN Y=Y+3
1470   IF D=3 THEN X=X+9
1480   IF D=5 THEN Y=Y-3
```

（1250～1300）飛行姿勢を90°ずつ左に回転させる

（1320～1370）飛行姿勢を90°ずつ右に回転させる

（1430～）ジョイスティックの動きに応じ，4方向に画面上移動表示させる

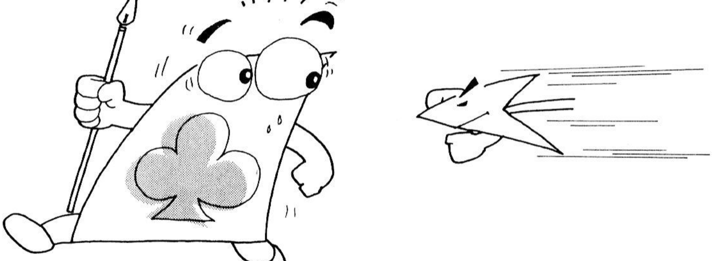
```
1490   IF D=7 THEN X=X-9
1500 PUT@A(X,Y)-(X+39,Y+7),JET1,PSET
1510 RETURN
1520 '
1530 '  ﾋﾀﾞﾘ 90ﾄﾞ ｶｲﾃﾝ ﾋｺｳ
1540 D=STICK(1)
1550 LINE(X,Y)-(X+39,Y+14),PSET,1,BF
1560   IF D=1 THEN X=X+9
1570   IF D=3 THEN Y=Y-3
1580   IF D=5 THEN X=X-9
1590   IF D=7 THEN Y=Y+3
1600 PUT@A(X,Y)-(X+19,Y+14),JET2,PSET
1610 RETURN
1620 '
1630 '  ﾊｲﾒﾝ ﾋｺｳ
1640 D=STICK(1)
1650 LINE(X,Y)-(X+39,Y+14),PSET,1,BF
1660   IF D=1 THEN Y=Y-3
1670   IF D=3 THEN X=X-9
1680   IF D=5 THEN Y=Y+3
1690   IF D=7 THEN X=X+9
1700 PUT@A(X,Y)-(X+39,Y+7),JET3,PSET
1710 RETURN
1720 '
1730 '  ﾐｷﾞ 90ﾄﾞ ｶｲﾃﾝ ﾋｺｳ
1740 D=STICK(1)
1750 LINE(X,Y)-(X+39,Y+14),PSET,1,BF
1760   IF D=1 THEN X=X-9
1770   IF D=3 THEN Y=Y+3
1780   IF D=5 THEN X=X+9
1790   IF D=7 THEN Y=Y-3
1800 PUT@A(X,Y)-(X+19,Y+14),JET4,PSET
1810 RETURN
1820 '
1830 '  ｶﾞﾒﾝ ﾉ ﾋｮｳｼﾞ ﾃﾞｰﾀ
1840 '
1850 '  ｽｲﾍｲ ﾋｺｳ ﾉ ｶﾞﾒﾝ ﾉ ﾋｮｳｼﾞ ﾃﾞｰﾀ
1860 DATA -1,-1,-1,-1,-1,-1,-1,-1
1870 DATA -1,-1,-1,-1,-1,-1,-1,-1
1880 DATA -1,-1,-1,-1,0,6144,0,2047
1890 DATA -8192,0,6144,0,24,0,0,15360
1900 DATA 192,511,-32765,-1,-1,-64,126,3
1910 DATA 0,6144,0,2047,-8192,0,6144,0
1920 DATA 24,0,0,15360,192,511,-32765,-1
1930 DATA -1,-64,126,3,0,0,0,0
1940 '
1950 '  ﾐｷﾞ 90ﾄﾞ ｶｲﾃﾝ ﾉ ｶﾞﾒﾝ ﾋｮｳｼﾞ ﾃﾞｰﾀ
1960 DATA -1,-1,-1,-1,-1,-1,-1,-1
1970 DATA -1,-1,-1,-1,-1,-1,-1,-1
1980 DATA -1,-1,-16,3,-4096,3072,192,12
1990 DATA 6144,-16000,15384,8191,-1,6175,
          -3712,15384
2000 DATA 192,12,0,-16384,3072,1008,3,-4096
```

(1540–1610: 同上; 1640–1710: 同上; 1740–1810: 同上; 1490–1810: 飛行位置制御)

```
2010 DATA 3072,192,12,6144,-16000,15384,8191,-1
2020 DATA 6175,-3712,15384,192,12,0,-16384,3072
2030 DATA 1008,0,0,0,0,0,0,0
2040 '
2050 '  ﾊｲﾒﾝ ﾋｺｳ ﾉ ｶﾞﾒﾝ ﾋｮｳｼﾞ ﾃﾞｰﾀ
2060 DATA -1,-1,-1,-1,-1,-1,-1,-1
2070 DATA -1,-1,-1,-1,-1,-1,-1,-1
2080 DATA -1,-1,-1,-1,-16384,32256,1023,-1
2090 DATA -1,-16383,-128,768,60,0,0,6144
2100 DATA 0,24,0,7,-32,0,24,0
2110 DATA -16384,32256,1023,-1,-1,-16383,
          -128,768
2120 DATA 60,0,0,6144,0,24,0,7
2130 DATA -32,0,24,0,0,0,0,0
2140 '
2150 '  ﾐｷﾞ 90ﾄﾞ ｶｲﾃﾝ ﾉ ｶﾞﾒﾝ ﾋｮｳｼﾞ ﾃﾞｰﾀ
2160 DATA -1,-1,-1,-1,-1,-1,-1,-1
2170 DATA -1,-1,-1,-1,-1,-1,-1,-1
2180 DATA -1,-1,-16,-1024,768,48,3,0
2190 DATA 12288,-15424,3327,-32561,-1,-128,
          -15424,3120
2200 DATA 195,0,12288,768,252,0,-1024,768
2210 DATA 48,3,0,12288,-15424,3327,-32561,-1
2220 DATA -128,-15424,3120,195,0,12288,768,252
```

8

FIRE（発射）と命中



8.1 水平接近と命中

ゲームでは，表示したパターンの移動とその衝突による命中表示という処理がしょっちゅう出てくる．

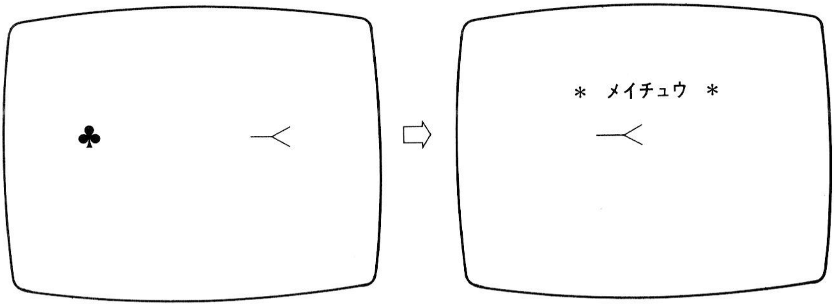
基本的には，接近する双方の表示位置を絶えず比べて，両者の値が同じになった時点で命中と判定し，命中の意示をするという方法をとる．

まず，簡単な例でみてみよう．つぎのプログラムは，人間に似せたクローバのマークと，投げ矢のようなものが左右から接近し，当ったところで命中を表示するものである．

```
10 '   P8.1  ｽｲﾍｲ ｾｯｷﾝ ﾄ ﾒｲﾁｭｳ
20 '
30 '
100 COLOR 7,1:WIDTH 40,25
110 X1=0:X2=38:Y=12
120  COLOR 7: LOCATE X1,Y:PRINT"♣";
130  COLOR 2: LOCATE X2,Y:PRINT"-<";
140   FOR T=1 TO 500:NEXT T
150 IF X1=X2 THEN 200
160 LOCATE X1,Y:PRINT" ";
170 LOCATE X2,Y:PRINT"  ";
180 X1=X1+1:X2=X2-1
190 GOTO 120
200 LOCATE 14,10 : COLOR 6
210 PRINT "*  ﾒｲﾁｭｳ !  *"
```

このプログラムでは，LOCATE 文により，人間と投げ矢の表示する場所を指定し，行180でＸ座標を変えることによって，両者の接近表示をしている．

行150で，両者のＸ座標（X1とX2）が等しくなったところで，命中と判定している．



8.2 垂直接近と命中

前の水平接近を，X座標を固定してY座標だけを動かすと，上下（垂直）移動となる．

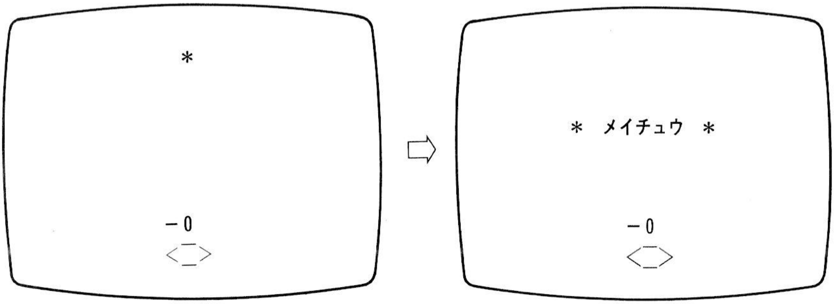
```
10 '   P8.2  ｽｲﾁｮｸ ｾｯｷﾝ ﾄ ﾒｲﾁｭｳ
20 '
30 '
100 COLOR 7,1:WIDTH 40,25
110 X=17:Y=0
120  COLOR 7: LOCATE X+2,Y:PRINT"*";
130  COLOR 2: LOCATE X,22:PRINT" -O";
140  COLOR 2: LOCATE X,23:PRINT"<‾‾‾>";
150  COLOR 2: LOCATE X,24:PRINT"  ‾‾‾";
160   FOR T=1 TO 500:NEXT T
170 IF Y=23 THEN 210
180 LOCATE X+2,Y:PRINT" ";
190 Y=Y+1
200 GOTO 120
210 LOCATE 13,10 : COLOR 6
220 PRINT "*  ﾒｲﾁｭｳ !  *"
```

例では，地上の戦車に上空から爆弾を落とし，爆撃するプログラムを示す．

爆弾である "＊" のパターンを画面の上から下へ動かすことによって，落下シーンを表示することができる．そのためには，"＊" パターン表示の場所を指定する **LOCATE** 文のY座標を0から増加させればよい（行190）．

このようにして，Yが戦車のある位置（Y＝23）のところまで

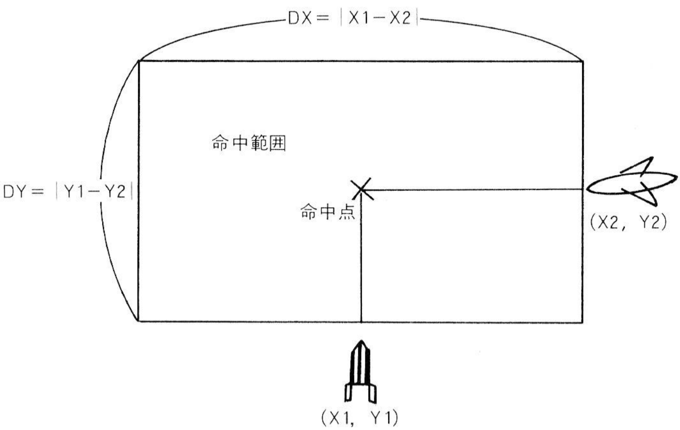
くれば命中としておけば，丁度爆弾が戦車の中にメリ込んだところで命中とすることができる．



8.3 水平・垂直接近と近接命中

水平接近と垂直接近をいっしょに合せて表示する例を示そう．ここでは，地上にあるミサイル戦車から上空を水平飛行する敵偵察機を撃墜するプログラムを例示してみる．ただし，ここで一寸工夫して，完全に命中しなくても，近くを通過しただけで，命中となるようにしてみた．

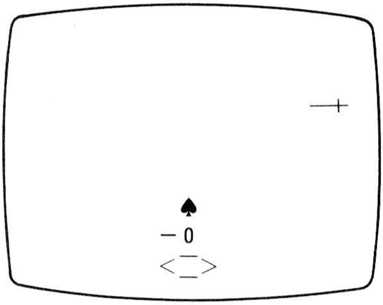
このプログラムでの要点は，行180～190の部分である．この2行で，近接命中の判定をしているのだ．

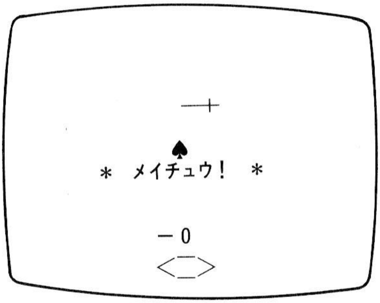


すなわち，ミサイルと偵察機の離隔距離を計算し，その絶対値が，ある一定の値以下となれば，その時点で命中と判定させている．絶対値としたのは，通過前であっても，通過後であっても，命中範囲であれば命中とするためである．

```
10 '   P8.3   キンセツ メイチュウ
20 '
30 '
100 COLOR 7,1:WIDTH 40,25
110 X1=12:Y1=21:X2=38:Y2=5
120  COLOR 2: LOCATE 10,22:PRINT" -O";
130  COLOR 2: LOCATE 10,23:PRINT"<¯¯¯>";
140  COLOR 2: LOCATE 10,24:PRINT" ¯¯¯";
150  COLOR 7: LOCATE X1,Y1:PRINT"♠";
160  COLOR 2: LOCATE X2,Y2:PRINT"-+";
170   FOR T=1 TO 500:NEXT T
180 DX=ABS(X1-X2):DY=ABS(Y1-Y2)
190 IF DX=<6 AND DY=<4 THEN 240
200 LOCATE X1,Y1:PRINT" ";
210 LOCATE X2,Y2:PRINT"  ";
220 X2=X2-3:Y1=Y1-2
230 GOTO 150
240 LOCATE 14,10 : COLOR 6
250 PRINT "*  メイチュウ !  *"
```

このプログラムを実行すると，つぎのように，2つのパターンが重なる前に，命中表示となることがわかる．






このへん
かな!!
ワ

9

サンプルゲーム

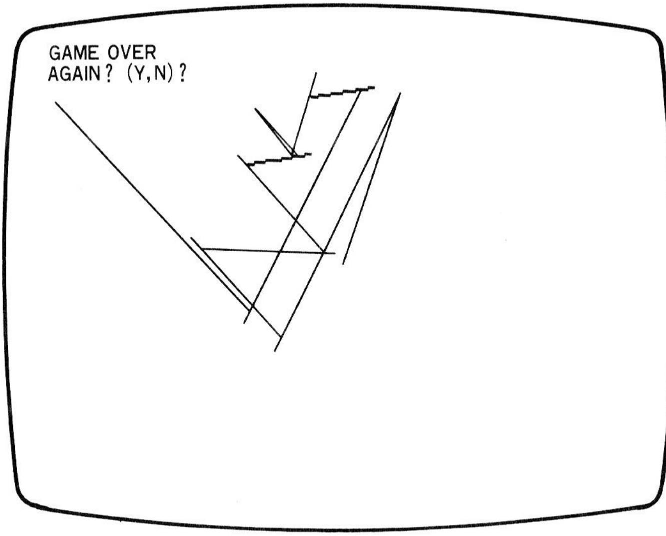


9.1 ロデオ・ゲーム

はじめに，ごく簡単なゲームから体験してみよう．このゲームは，線を引く LINE 文を主体に，進む方向を乱数で角度を決め，三角関数で，座標位置に直すという手法を用いたものである．外周に枠を設定し，枠に外れるとゲーム・オーバーにするというルールをつけた．

実行後，スペース・キーを押して方向を変えるが，どちらに進むのかは全くわからない．画面の右上にポイントが表示されるが，このポイントをできるだけ多くした方が勝ちというわけである．

あたかも，荒れ馬にひきずられ，どれだけ落馬しないで頑張っていられるかの，あのロデオにちなんで，ロデオ・ゲームと名づけてみた．

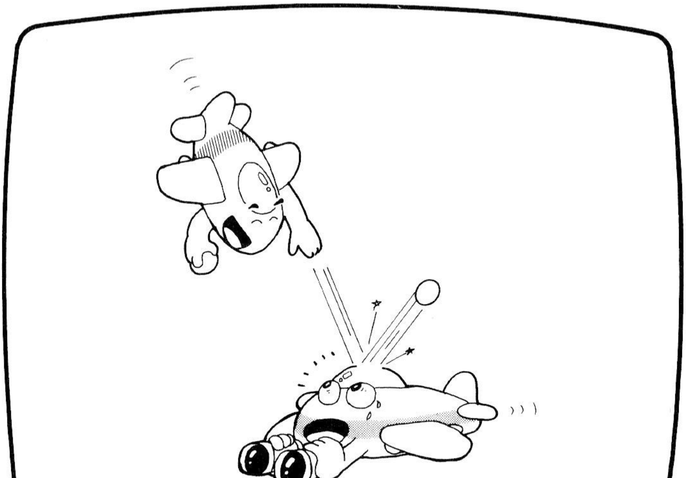



```
10 ' GM1 RODEO
20 '
30 '
100 WIDTH 40
105 LOCATE 7,12:PRINT"<ｽﾍﾟｰｽ> ｷｰ ｦ ｵｽﾄ, ｽﾀｰﾄ ｼﾏｽ";
110 A$=INPUT$(1)
120 FOR T=1 TO 500:NEXT T:P=0
130 LINE(0,0)-(639,399),PSET,6,BF
140 X=320 :Y=100
150 TH=RND*6.28
160 DX=10*COS(TH):DY=15*SIN(TH)
170 LINE(X,Y)-(X+DX,Y+DY),PSET,2
180 LOCATE 30,0:PRINT"POINT:";P;
190 P=P+1
200 FOR T=1 TO 50:NEXT T
210 X=X+DX:Y=Y+DY
240 IF X=>639 OR X<=0 THEN 280
250 IF Y=>199 OR Y<=0 THEN 280
260 K$=INKEY$
270 IFK$=" " THEN 150 ELSE 170
280 LOCATE 5,24:PRINT"<ｽﾍﾟｰｽ> ｷｰ  ---->  ﾂｷﾞﾆ ｽｽﾑ !";
285 K$=INPUT$(1)
290 COLOR 1,4:WIDTH 40
300 LOCATE 10,10:PRINT"G A M E  O V E R"
310 LOCATE 11,14:INPUT "AGAIN? (Y,N)";R$
320 IF R$="Y" THEN 100
330 IF R$="N" THEN 340
340 COLOR 7,0:WIDTH 80
```

9.2 空中戦

まえのプログラムをベースに，これを二人でできるようにすると，空中戦のゲームになる．

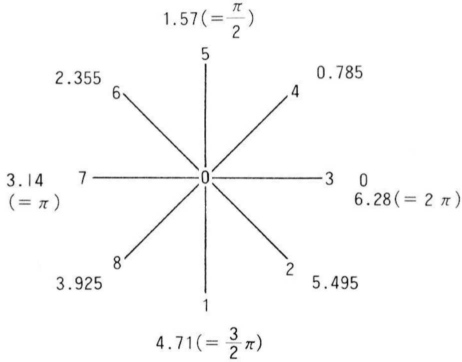
ジョイスティックを2本使い，どちらかが逃げ，それを他方が追うというゲーム．できるだけ長く逃げ回ることができた方が勝ち．攻・守交代しながらやってみると，結構面白くなる．



```
10 ' GM2 SPACE FIGHT
20 '
30 '
100 WIDTH 40
110 LOCATE 7,12
120 PRINT"<ｽﾍﾟｰｽ> ｷｰ ｦ ｵｽﾄ, ｽﾀｰﾄ ｼﾏｽ";
130 A$=INPUT$(1)
140 FOR T=1 TO 500:NEXT T:P=0
150 LINE(0,0)-(639,399),PSET,6,BF
160 XA=200:YA=100 : XB=400:YB=100
170 DX1=21*COS(4.71) :DY1=7*SIN(4.71)
180 DX2=21*COS(5.495):DY2=7*SIN(5.495)
190 DX3=21*COS(0)     :DY3=7*SIN(0)
200 DX4=21*COS(.785) :DY4=7*SIN(.785)
210 DX5=21*COS(1.57) :DY5=7*SIN(1.57)
220 DX6=21*COS(2.355):DY6=7*SIN(2.355)
230 DX7=21*COS(3.14) :DY7=7*SIN(3.14)
240 DX8=21*COS(3.925):DY8=7*SIN(3.925)
250 '
260 LOCATE 30,0:PRINT"POINT:";P;
270 P=P+1
280 KA=STICK(1):KB=STICK(2)
290 IF KA=1 THEN DXA=DX1:DYA=DY1
300 IF KA=2 THEN DXA=DX2:DYA=DY2
310 IF KA=3 THEN DXA=DX3:DYA=DY3
320 IF KA=4 THEN DXA=DX4:DYA=DY4
330 IF KA=5 THEN DXA=DX5:DYA=DY5
340 IF KA=6 THEN DXA=DX6:DYA=DY6
350 IF KA=7 THEN DXA=DX7:DYA=DY7
360 IF KA=8 THEN DXA=DX8:DYA=DY8
370 IF KB=1 THEN DXB=DX1:DYB=DY1
380 IF KB=2 THEN DXB=DX2:DYB=DY2
390 IF KB=3 THEN DXB=DX3:DYB=DY3
400 IF KB=4 THEN DXB=DX4:DYB=DY4
410 IF KB=5 THEN DXB=DX5:DYB=DY5
420 IF KB=6 THEN DXB=DX6:DYB=DY6
430 IF KB=7 THEN DXB=DX7:DYB=DY7
440 IF KB=8 THEN DXB=DX8:DYB=DY8
450 IF XA=>639 OR XA<=0 THEN 540
460 IF YA=>199 OR YA<=0 THEN 540
470 IF XB=>639 OR XB<=0 THEN 540
480 IF YB=>199 OR YB<=0 THEN 540
490 LINE(XA,YA)-(XA+DXA,YA+DYA),PSET,2
500 LINE(XB,YB)-(XB+DXB,YB+DYB),PSET,1
510 IF ((XB-21)<=XA AND XA<=(XB+21)) AND
       ((YB-7)<=YA AND YA<=(YB+7)) THEN 540
520 XA=XA+DXA:YA=YA+DYA:XB=XB+DXB:YB=YB+DYB
530 GOTO 170
540 LOCATE 5,24
550 PRINT"<ｽﾍﾟｰｽ> ｷｰ  ---->  ﾂｷﾞﾆ ｽｽﾑ !";
560 K$=INPUT$(1)
570 COLOR 1,4:WIDTH 40
580 LOCATE 10,10:PRINT"G A M E  O V E R"
```

```
590 LOCATE 11,14:INPUT "AGAIN? (Y,N)";R$
600 IF R$="Y" THEN 100
610 IF R$="N" THEN 620
620 COLOR 7,0:WIDTH 80
```

ジョイスティックの方向と対応する角度はつぎのような関係になっている．内側の１～８の数は，ジョイスティックがその方向に傾いたときに，**STICK（　）**関数に与えられる数で，外形の小数は，角度をラジアンで表示した値である．



以上のジョイスティックの位置関係から，行170～240で，それぞれの方向にジョイスティックが倒された場合の線の進み幅（**DX1～DX8，DY1～DY8**）を求めている．

ついで，行290～440に，ジョイスティックの倒された方向により，上で求めた各方向に応じた線の進み幅を選択するための IF 文を並べてある．

このようにIF文を並べることによって，ジョイスティックの指示によって，線の進む方向を8方向のなかから，自由に選ぶことができるようにすることが可能となる．

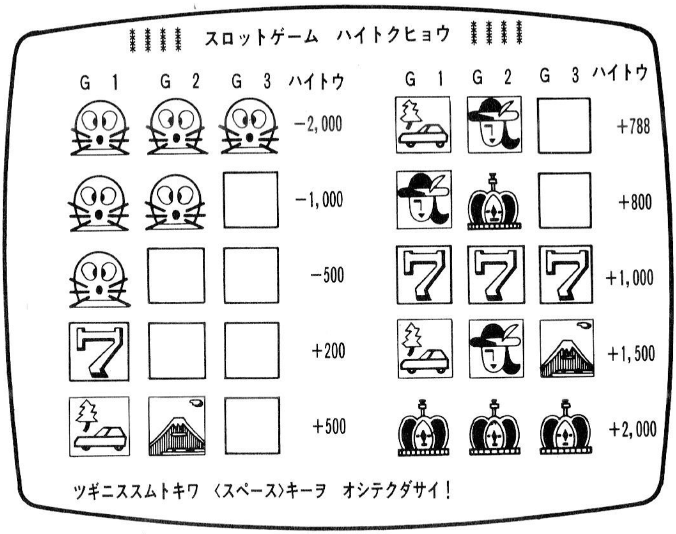
9.3 スロット・ゲーム

最後に少し長くなるが，スロット・ゲームを，きれいな絵を楽しみながら体験してみよう．

最後に掲載した500ステップ余りのプログラムを入力することからはじめる．

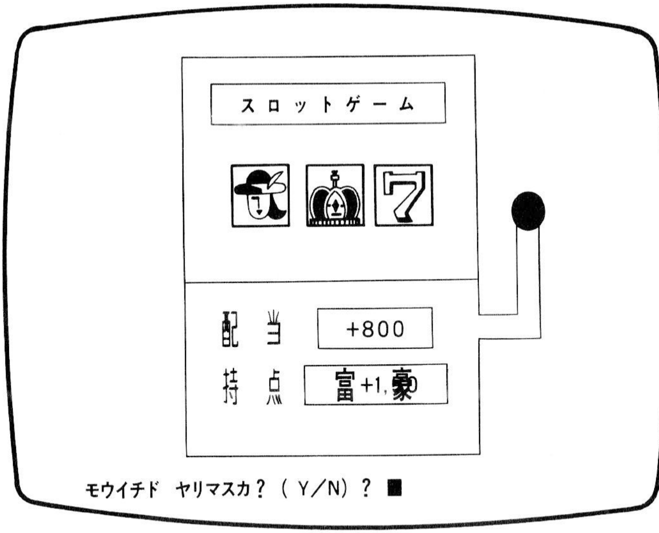
実行は簡単で，画面上に出てくる指示通り進め，あとは，スペース・キーを押していくだけである．

このプログラムを実行すると，まず，大量の絵のデータを配列変数に読み込むため，やや時間（7～8秒）がかかるが，それが終ると，つぎのような美しいパターン表示による，ゲームの配当表が出てくる．この絵の並び方で配当が変るのが，スロット・ゲームであるが，一，二枚の絵の組み合せの場合は，それがどの位置に，どういう順序で並んでいてもよいことにしてある．また，三枚組みの場合も，どういう順序に出てきてもOKとしている．



つぎに，このゲームをスタートする前にランダムな数を作るための基礎の数を指定することになっている．このステップを設けることによって，ランダムな数を発生させるルールを変えることができるので，いつでも違った絵の出方にすることができる．

さて，いよいよゲーム開始．つぎのようなスロット・マシーンの絵が出てくる．



ルールはつぎのようになっている．

持ち点	1000ポイント		
ゲーム1回毎にゲーム代をして	200	〃	引かれる
破　産	0	〃	以下
富　豪	2000	〃	以上

①スペース・キーを押すと絵が回転を始める．

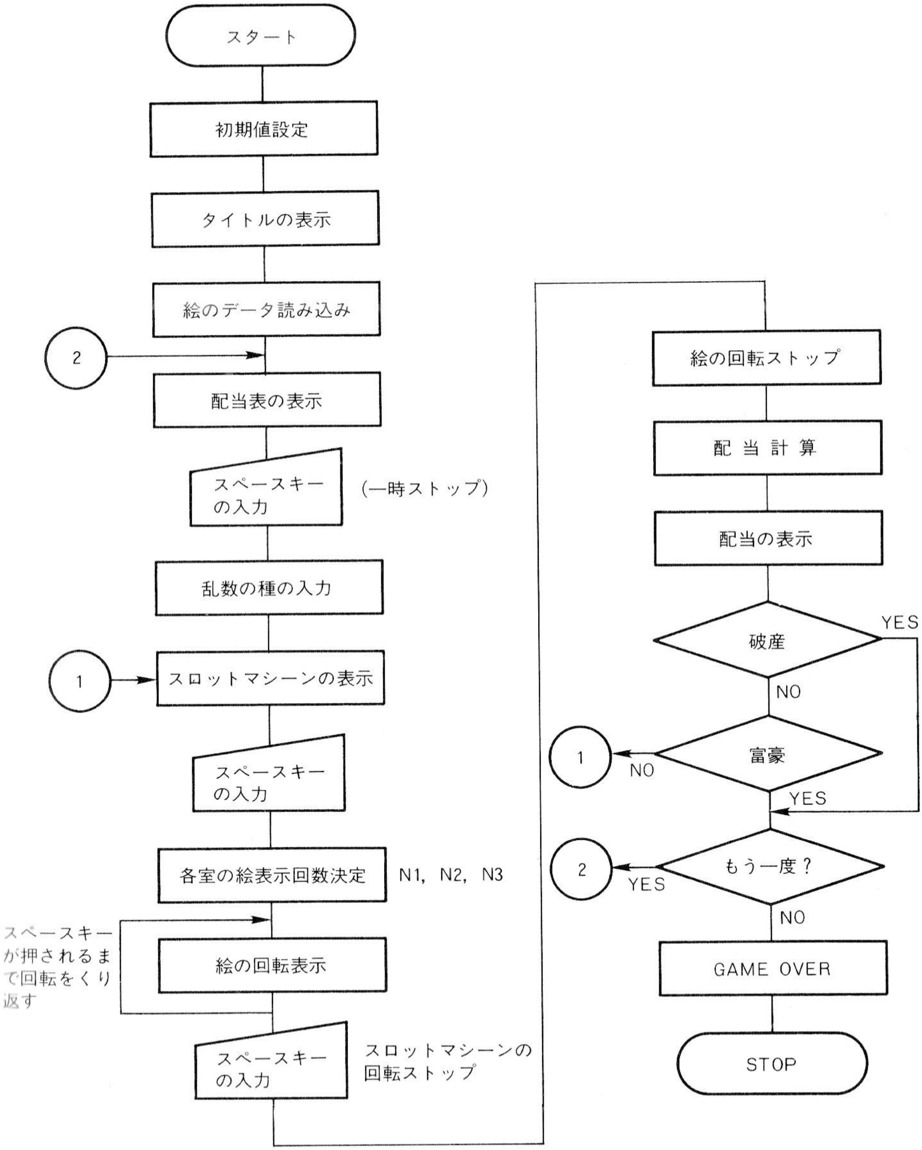
②もう一度スペース・キーを押すと，窓毎に，絵が止まり始める．完全に止ったところで配当が決まる．

③つぎにスペース・キーを押すと，配当は0にもどり，持点を計算表示する．

持点＝持点＋配当－200

以上をくり返す．

スロットゲームのフロー



```
10 ' GM3   ｽﾛｯﾄ･ｹﾞｰﾑ
20 '
30 '
1000 SCREEN@ 0:COLOR 6,1:DEFINT A-Z:WIDTH 40,25
1010 DIM PAT1(543),PAT2(543),PAT3(543),
        PAT4(543),PAT5(543),PAT6(543)
1020 P1=1:P2=2:P3=3:P4=4:P5=5:P6=6
1030 THAITO!=2000!
1040 '
1050 '  ﾀｲﾄﾙ ｶﾞﾒﾝ ﾉ ﾋｮｳｼﾞ
1060 LOCATE 6,8:PRINT"<<<   S L O T   G A M E   >>>"
1070 LOCATE 9,14:PRINT"ｲﾏ, ﾃﾞｰﾀ ｦ ﾖﾐｺﾝﾃﾞ ｲﾏｽ｡"
1080 '
1090 '  ﾊﾟﾀｰﾝ･ﾃﾞｰﾀ ﾉ ﾖﾐｺﾐ
1100 'ﾓｸﾞﾗ
1110 FOR I=0 TO 271
1120   READ PAT1(I):NEXT I
1130 '  CAR
1140 FOR I=0 TO 271
1150 READ PAT2(I):NEXT I
1160 '  GAL
1170 FOR I=0 TO 263
1180 READ PAT3(I):NEXT I
1190 '  Mt. FUJI
1200 FOR I=0 TO 271
1210   READ PAT4(I):NEXT I
1220 '  ﾗｯｷｰ ｾﾌﾞﾝ
1230 FOR I=0 TO 271
1240   READ PAT5(I):NEXT I
1250 '  ｸﾗｳﾝ
1260 FOR I=0 TO 271
1270   READ PAT6(I):NEXT I
1280 '
1290 '  ﾊｲﾄｳﾋｮｳ ﾉ ﾋｮｳｼﾞ
1300 WIDTH 80:LOCATE 11,0:COLOR 6
1310 PRINT"* * * *   ｽ ﾛ ｯ ﾄ ﾙ ･ ｹ ﾞ ｰ ﾑ   ﾊ ｲ ﾄ ｳ ﾋ ｮ ｳ   * * * *"
1320 LOCATE 2,2:COLOR 5
1330 PRINT"G  1     G  2     G  3     ﾊｲﾄｳ";:LOCATE 46,2
1340 PRINT"G  1     G  2     G  3     ﾊｲﾄｳ":COLOR 7
1350 '
1360      PUT@A(0,32)-(0+59,32+23),PAT1,PSET
1370      PUT@A(80,32)-(80+59,32+23),PAT1,PSET
1380      PUT@A(160,32)-(160+59,32+23),PAT1,PSET
1390      PUT@A(352,32)-(352+59,32+23),PAT2,PSET
1400      PUT@A(432,32)-(432+59,32+23),PAT3,PSET
1410      PUT@A(0,64)-(0+59,64+23),PAT1,PSET
1420      PUT@A(80,64)-(80+59,64+23),PAT1,PSET
1430      PUT@A(352,64)-(352+59,64+23),PAT3,PSET
1440      PUT@A(432,64)-(432+59,64+23),PAT6,PSET
1450      PUT@A(0,96)-(0+59,96+23),PAT5,PSET
1460      PUT@A(352,96)-(352+59,96+23),PAT5,PSET
1470      PUT@A(432,96)-(432+59,96+23),PAT5,PSET
1480      PUT@A(512,96)-(512+59,96+23),PAT5,PSET
1490      PUT@A(0,128)-(0+59,128+23),PAT5,PSET
1500      PUT@A(352,128)-(352+59,128+23),PAT2,PSET
1510      PUT@A(432,128)-(432+59,128+23),PAT3,PSET
1520      PUT@A(512,128)-(512+59,128+23),PAT4,PSET
1530      PUT@A(0,160)-(0+59,160+23),PAT2,PSET
1540      PUT@A(80,160)-(80+59,160+23),PAT4,PSET
1550      PUT@A(352,160)-(352+59,160+23),PAT6,PSET
1560      PUT@A(432,160)-(432+59,160+23),PAT6,PSET
1570      PUT@A(512,160)-(512+59,160+23),PAT6,PSET
1580    FOR I=1 TO 8
1590       READ X,Y:LINE(X,Y)-(X+59,Y+23),PSET,7,B
```

```
1600     NEXT I
1610 '
1620 FOR I=1 TO 5
1630   READ Y,R1,R2
1640   LOCATE 30,Y:PRINT USING"+#,###";R1
1650   LOCATE 74,Y:PRINT USING"+#,###";R2
1660 NEXT I
1670 '
1680 LOCATE 15,24:COLOR 4
1690 PRINT"ツギニススムトキハ, <スペース>キー ヲ オシテ クダサイ !";:COLOR 7
1700   NXT$=INPUT$(1)
1710 '
1720 COLOR 7,1 : WIDTH 40
1730 COLOR 6:LOCATE 10,3:PRINT"レ ッ ツ  ス タ ー ト !!":COLOR 7
1740 LOCATE 2,7:PRINT"シタ ノ スウジ ハ ゲーム ノ ウン ヲ キメル カズデス"
1750 COLOR 6
1760 LOCATE 9,10:PRINT"スキナカズ ヲ ニュウリョク シテクダサイ"
1770 LOCATE 0,16
1780 RANDOMIZE
1790 '
1800 ' スロット・ゲーム ガメン ノ ヒョウジ
1810 CLS
1820 LINE(160,16)-(480,112),PSET,3,BF
1830 LINE(192,28)-(448,44),PSET,6,BF
1840 COLOR 7,6
1850 LOCATE 13,4:PRINT"ス ロ ッ ト ゲ ー ム"
1860 LINE(211,63)-(272,88),PSET,7,B
1870 LINE(295,63)-(356,88),PSET,7,B
1880 LINE(371,63)-(432,88),PSET,7,B
1890 LINE(160,113)-(480,184),PSET,7,BF
1900 COLOR 0,1
1910 PRINT@(200,124),&H475B,&H3020,&H3020,&H4576
1920 COLOR 2
1930 PRINT@(200,148),&H3B7D,&H3020,&H3020,&H4540
1940 LINE(304,124)-(432,140),PSET,1,BF
1950 LINE(288,148)-(448,164),PSET,1,BF
1960 LINE(480,127)-(552,136),PSET,7,BF
1970 LINE(528,88)-(552,136),PSET,7,BF
1980 CIRCLE(541,82),20,2,,,,F,PSET
1990 COLOR 7
2000 X=212:Y=64:GOSUB 2810
2010 X=296:Y=64:GOSUB 2810
2020 X=372:Y=64:GOSUB 2810
2030 '
2040 LOCATE 0,23:COLOR 6
2050 LOCATE 20,16:PRINT USING"+#,###";HAITO;
2060 LOCATE 19,19:PRINT USING"+###,###";THAITO!;
2070 LOCATE 0,24
2080 PRINT"<スペース>キー デ スタート/ストップ !";:COLOR 7
2090   NXT$=INPUT$(1)
2100 '  スロット・マシーン スタート
2110 WK1$=""
2120 N1=INT(20*RND)+1:N2=INT(20*RND)+1:N3=INT(20*RND)+1
2130 GOSUB 2700
2140  WK1$=INKEY$
2150  IF WK1$=" " THEN 2180 ELSE 2110
2160 '
2170 '  ストップ ルーチン
2180 L1=0:L2=0:L3=0
2190 IF L1=N1 THEN 2250
2200   X=212:Y=64
2210   G1=(L1 MOD 6)+1
2220     ON G1 GOSUB 2810,2840,2870,2900,2930,2960
2230   L1=L1+1
```

```
2240 '
2250 IF L2=N2 THEN 2310
2260   X=296:Y=64
2270   G2=(L2 MOD 6)+1
2280     ON G2 GOSUB 2810,2840,2870,2900,2930,2960
2290   L2=L2+1
2300 '
2310 IF L3=N3 THEN 2370
2320   X=372:Y=64
2330   G3=(L3 MOD 6)+1
2340     ON G3 GOSUB 2810,2840,2870,2900,2930,2960
2350   L3=L3+1
2360 GOTO 2190
2370 '
2380 LOT=100*G1+10*G2+G3
2390 IF LOT=111 THEN HAITO=-2000:GOTO 2490
2400 IF LOT=555 THEN HAITO=700:GOTO 2490
2410 IF LOT=666 THEN HAITO=2000:GOTO 2490
2420 IF LOT=234 OR LOT=243 OR LOT=324 OR LOT=342 OR
     LOT=423 OR LOT=432 THEN HAITO=1000:GOTO 2490
2430 IF (G1=1 AND G2=1) OR (G2=1 AND G3=1) OR
        (G1=1 AND G3=1) THEN HAITO=-1000 :GOTO 2490
2440 IF (G1=2 AND (G2=4 OR G3=4)) OR (G1=4 AND (G2=2 OR G3=2))
     OR (G2=2 AND G3=4) OR (G2=4 AND G3=2) THEN HAITO=500:GOTO 2490
2450 IF (G1=2 AND (G2=3 OR G3=3)) OR (G1=3 AND (G2=2 OR G3=2))
     OR (G2=2 AND G3=3) OR (G2=3 AND G3=2) THEN HAITO=500:GOTO 2490
2460 IF (G1=3 AND (G2=6 OR G3=6)) OR (G1=6 AND (G2=3 OR G3=3))
     OR (G2=3 AND G3=6) OR (G2=6 AND G3=3) THEN HAITO=500:GOTO 2490
2470 IF G1=5 OR G2=5 OR G3=5 THEN HAITO=200 :GOTO 2490
2480 IF G1=1 OR G2=1 OR G3=1 THEN HAITO=-500:GOTO 2490
2490 '
2500 THAITO!=THAITO!+HAITO
2510 THAITO!=THAITO!-200
2520 IF THAITO!<-1000 THEN COLOR 2:PRINT@(312,148),&H3020,
     &H474B,&H3020,&H3020,&H3020,&H3B3A:GOTO 2610
2530 IF THAITO!>5000 THEN COLOR 6:PRINT@(312,148),&H3020,
     &H4959,&H3020,&H3020,&H3020,&H396B:GOTO 2610
2540 COLOR 2
2550 LOCATE 20,16:PRINT USING"+#,###";HAITO;:HAITO=0
2560 LOCATE 19,19:PRINT USING"+###,###";THAITO!;
2570 '
2580 NXT$=INPUT$(1)
2590 GOTO 1990
2600 '
2610 COLOR 6:LOCATE 0,24
2620 INPUT"ﾓｳｲﾁﾄﾞ ﾔﾘﾏｽｶ? ( Y / N )";K$
2630 IF K$="Y" OR K$="y" THEN WIDTH 80:RESTORE 5120:
     HAITO=0:THAITO!=2000:GOTO 1290
2640 IF K$="N" OR K$="n" THEN CLS ELSE 2610
2650 LOCATE 4,12:PRINT"* * *  G A M E   O V E R  * * *"
2660 '
2670 END
2680 '
2690 '  ﾊﾟﾀｰﾝ ｶｲﾃﾝ ﾉ ｻﾌﾞﾙｰﾁﾝ
2700 FOR N=1 TO 6
2710   X=212:Y=64
2720     ON N GOSUB 2810,2840,2870,2900,2930,2960
2730  X=296:Y=64
2740     ON N GOSUB 2810,2840,2870,2900,2930,2960
2750  X=372:Y=64
2760     ON N GOSUB 2810,2840,2870,2900,2930,2960
2770  NEXT N
2780 RETURN
2790 '
```

```
2800 'ﾊﾟﾀｰﾝ ﾋｮｳｼﾞ ﾉ ｻﾌﾞﾙｰﾁﾝ
2810 PUT@A(X,Y)-(X+59,Y+23),PAT1,PSET
2820 RETURN
2830 '
2840 PUT@A(X,Y)-(X+59,Y+23),PAT2,PSET
2850 RETURN
2860 '
2870 PUT@A(X,Y)-(X+59,Y+23),PAT3,PSET
2880 RETURN
2890 '
2900 PUT@A(X,Y)-(X+59,Y+23),PAT4,PSET
2910 RETURN
2920 '
2930 PUT@A(X,Y)-(X+59,Y+23),PAT5,PSET
2940 RETURN
2950 '
2960 PUT@A(X,Y)-(X+59,Y+23),PAT6,PSET
2970 RETURN
2980 '
2990 '
3000 '   ﾊﾟﾀｰﾝ ﾉ ﾃﾞｰﾀ
3010 '
3020 ' ﾓｸﾞﾗ
3030 DATA -1,-1,-1,-1,-1,0,4095,-1
3040 DATA -1024,0,1023,-2,0,0,2047,-128
3050 DATA 0,0,8191,-4096,0,0,-4,0
3060 DATA 0,3,-128,0,0,31,-4096,0
3070 DATA 0,254,0,3591,0,2016,3,-3844
3080 DATA 0,32256,127,4064,7,-8192,992,31744
3090 DATA 112,0,3075,0,224,0,0,0
3100 DATA 32512,0,0,15,-4096,0,0,255
3110 DATA -32768,0,0,7936,0,0,0,4064
3120 DATA 0,0,127,-256,0,0,4095,-2048
3130 DATA 0,1,-256,0,0,0,0,0
3140 DATA 0,0,0,0,0,0,0,0
3150 DATA 0,0,15,-1,0,0,8191,-1
3160 DATA -32768,15,-1,-1,0,1023,-1,-4
3170 DATA 0,-128,-16,8176,31,-16389,-513,-8320
3180 DATA 1015,-33,-16385,31744,32639,-3592,-9,-8185
3190 DATA -2052,3843,-130,127,32640,-4065,-2080,2044
3200 DATA -993,-31751,-512,32764,1020,1023,-8192,32767
3210 DATA -1,-64,56,16383,-1,960,1020,4095
3220 DATA -497,-1024,8191,-1,-1,-32768,7,-2
3230 DATA 0,7,-1,-1,-512,60,2047,-497
3240 DATA -16384,31,-1,-512,0,0,0,0
3250 DATA 0,0,0,0,0,0,0,0
3260 DATA 0,0,0,0,15,-1,0,0
3270 DATA 8191,-1,-32768,15,-1,-1,0,1023
3280 DATA -1,-4,0,-128,-16,8176,31,-16389
3290 DATA -513,-8320,1015,-33,-16385,31744,32639,-3592
3300 DATA -9,-8185,-2052,3843,-130,127,32640,-4065
3310 DATA -2080,2044,-993,-31751,-512,32764,1020,1023
3320 DATA -8192,32767,-1,-64,56,16383,-1,960
3330 DATA 1020,4064,32271,-1024,8191,-1021,-1,-32768
3340 DATA 7,-16322,0,7,-2,2047,-512,60
3350 DATA 2047,-497,-16384,31,-1,-512,248,8191
3360 DATA -1,-64,-3969,-1,-1,-16,0,0
3370 '  CAR
3380 DATA 0,0,0,0,0,0,0,0
3390 DATA 3072,0,0,0,0,0,0,0
3400 DATA 0,0,0,0,0,0,0,30720
3410 DATA 0,0,0,0,0,0,0,30720
3420 DATA 0,0,0,0,0,0,0,0
3430 DATA 0,0,0,0,0,0,0,32256
```

```
3440 DATA 0,0,63,-28688,0,0,16323,-64
3450 DATA 0,15,-3841,-256,0,0,0,0
3460 DATA 0,0,0,0,0,0,0,0
3470 DATA 0,0,0,7,-32768,1,-8192,0
3480 DATA 0,0,0,0,0,0,0,0
3490 DATA 0,0,-1,-1,-1,-1,-127,-1
3500 DATA -1,-1,-8177,-1,-1,-8,15,-1
3510 DATA -1,-4,1023,-1,-1,-256,32767,-1
3520 DATA -1,-32768,-1,-1,-4,3,-1,-1
3530 DATA -512,127,-1,-1,-32768,1023,-1,-1
3540 DATA 0,4095,-1,-1,-1,-1,-1,-1
3550 DATA -1,-32257,-1,-1,-64,28687,-1,-1
3560 DATA -16324,63,-1,-16,3840,255,-1,-1
3570 DATA -1,-1,-1,-1,-1,-1,-1,-1
3580 DATA -1,-1,-1,-1,-1,-57,-28673,-31
3590 DATA -7681,-253,-1,-128,32767,-1,-1,-1
3600 DATA -1,-1,-1,-1,-1,-1,-1,-1
3610 DATA -1,-1,-1,-1,-3073,-1,-1,-1
3620 DATA -1,-1,-1,-1,-1,-1,-1,-1
3630 DATA -1,-1,-1,-30721,-1,-1,-1,-1
3640 DATA -1,-1,-1,-30721,-1,-1,-1,-1
3650 DATA -1,-1,-1,-1,-1,-1,-15873,-1
3660 DATA -1,-4,8191,-32768,8191,-125,-64,0
3670 DATA -1,-8065,-16384,3,-1,-16,0,7
3680 DATA -256,0,0,62,0,0,0,384
3690 DATA 0,0,0,15360,0,0,7,-57
3700 DATA -28673,-31,-7681,-253,-1,-128,32767,-1
3710 DATA -1,-1,-1,-1,-1,-1,0,0
3720 '  GAL
3730 DATA -1,-1,-1,-1,-1,-16383,-1,8191
3740 DATA -32,7,-1017,-1,-4096,14,511,-4
3750 DATA 0,0,-1,-128,0,511,-1,-2048
3760 DATA 0,8191,-1024,16383,-499,-1,0,8191
3770 DATA -3617,-4,0,15,-1,-16,0,0
3780 DATA 4095,-1,-16384,0,32767,-4,0,7
3790 DATA -1,-64,0,127,-1,-1024,0,1023
3800 DATA -1,-16384,0,8191,-2,0,0,-1
3810 DATA -16,0,3,-6145,-128,0,0,16383
3820 DATA -256,96,7,-1,-512,15872,511,-1
3830 DATA -1,-4096,-1,-1,-1,-1,-1,-1
3840 DATA -1,-1,-1,-1,-1,-1,-1,-16383
3850 DATA -8,8191,-32,7,-1017,-1,-4096,14
3860 DATA 511,-4,0,0,-1,-128,0,511
3870 DATA -1,-2048,14,8191,-1024,16383,-3,-1
3880 DATA 0,8191,-33,-4,0,15,-1,-16
3890 DATA 0,511,-4097,-1,-1,-2,32767,-1
3900 DATA -16369,-25,-1,-1,-257,-385,-1,-1
3910 DATA -4097,-7169,-1,-2,-2,8191,-1,-17
3920 DATA -64,-1,-1,-1,-2045,-6145,-1,-1
3930 DATA 0,16383,-1,-32,7,-1,-1,-512
3940 DATA 511,-1,-1,-4096,-1,-1,-1,-1
3950 DATA -1,-1,-1,-1,0,0,0,0
3960 DATA 0,0,7,-8192,0,0,1016,0
3970 DATA 0,1,-512,0,0,127,0,0
3980 DATA 0,3584,0,0,1,-32768,0,0
3990 DATA 0,16368,0,0,15,-64,0,15
4000 DATA -1,-4096,0,511,-8192,0,16383,-2
4010 DATA 0,3,-16369,-32,0,63,-257,-512
4020 DATA 0,1023,-4097,-8192,0,16382,-2,0
4030 DATA 1,-17,-64,0,15,-1,-2048,0
4040 DATA 127,127,0,0,254,16256,0,0
4050 DATA 511,-16384,0,0,0,0,0,0
4060 '  Mt. FUJI
4070 DATA -1,-1,-1,-1,-1,-1,-1,-1
```

```
4080 DATA -1,-1,-1,-1,-1,-1,-1,-1
4090 DATA -1,-1,-1,-1,-1,-1,-1,-1
4100 DATA -1,-1,-1,-1,-1,-1,-1,-1
4110 DATA -1,-1,-1,-1,-1,-1,-1,-1
4120 DATA -1,-1,-1,-1,-1,-1,-1,-1
4130 DATA -1,-1,-1,-1,-1,-1,-1,-1
4140 DATA -1,-4096,0,-1,-16,4095,-4096,16383
4150 DATA -2048,2047,-1024,32764,0,0,0,-512
4160 DATA 0,0,0,0,0,0,0,0
4170 DATA 0,0,3,-1,-1,-1,-8192,-1
4180 DATA -1,-1,0,0,0,0,0,0
4190 DATA 1023,-32768,0,0,1023,0,0,0
4200 DATA 0,0,0,0,0,0,0,0
4210 DATA 0,0,0,0,0,0,0,0
4220 DATA 0,51,12288,0,0,4095,-16384,0
4230 DATA 3,-1,0,0,248,-13188,0,0
4240 DATA 15360,240,0,14,0,448,0,768
4250 DATA 0,768,0,-16384,0,3072,112,0
4260 DATA 0,14336,14336,0,0,28700,0,0
4270 DATA 0,-4608,0,0,0,0,0,0
4280 DATA 0,15,-1,-1,-256,0,0,0
4290 DATA 0,0,0,0,-1,-1,-1,-1
4300 DATA -1,-1,-1,-1,-1,-1,-3073,32767
4310 DATA -1,-1,-511,-1,-1,-1,-1,-1
4320 DATA -1,-1,-1,-1,-1,-1,-1,-1
4330 DATA -1,-1,-1,-1,-1,-1,-1,-1
4340 DATA -1,-1,-1,-1,-1,-1,-8,-13185
4350 DATA -1,-1,-1024,255,-1,-2,0,511
4360 DATA -1,-256,-2,1023,-1,-12289,-1,4095
4370 DATA -1,-4096,4095,-1,-1,-2048,1023,-1
4380 DATA -1,-1,-1,-1,-1,-1,-1,-1
4390 DATA -1,-1,-1,-1,-7169,-16369,-4,0
4400 DATA 0,0,8191,0,0,0,0,0
4410 '  ﾗｯｷｰ ｾﾌﾞﾝ
4420 DATA -1,-1,-1,-1,-1,-1,-1,-2
4430 DATA 1023,-1,-2017,-32,0,0,511,-512
4440 DATA 0,0,8191,-8192,0,1,-2,0
4450 DATA 0,31,-32,16383,-4,511,-509,-1
4460 DATA -64,8191,-8129,-1,-4093,-2,1023,-4
4470 DATA 255,-1,-1,-256,16383,-1,-1,-32753
4480 DATA -1,-1,-64,1023,-1,-1,-8192,-1
4490 DATA -1,-16,63,-1,-1,-1024,4095,-1
4500 DATA -1,1,-1,-1,-64,63,-1,-1
4510 DATA -1024,2047,-1,-1,-8192,16383,-1,-1
4520 DATA 1,-1,-1,-8,15,-1,-1,-1
4530 DATA -1,-1,-1,-1,-1,-1,-1,-1
4540 DATA -1,-1,-897,-1,-29,-1,-1,-1
4550 DATA -449,-1,-1,-1,-7169,-1,-1,-2
4560 DATA 16383,-1,-1,-29,-1,-16384,3,-449
4570 DATA -4,32767,-1,-7169,-57,-1,-4,32767
4580 DATA -897,-1,-225,-32,2047,-1,-14337,-1
4590 DATA -1,-15,-1,-1,-1,-897,-1,-1
4600 DATA -1,8191,-1,-1,-57,-1,-1,-1
4610 DATA -3585,-1,-1,-2,16383,-1,-1,-57
4620 DATA -1,-1,-1,-1793,-1,-1,-1,-14337
4630 DATA -1,-1,-2,16383,-1,-1,-15,-1
4640 DATA -1,-64,15,-1,-1,-1,-1,-1
4650 DATA -1,-1,-1,-2,127,-1,-2045,-25
4660 DATA 0,0,63,-385,-1,-1,-31745,-6145
4670 DATA -1,-8,16382,28672,0,1923,-25,0
4680 DATA 0,30783,-400,32767,-57,-31745,-6393,-1
4690 DATA -3616,32766,127,-4,30751,-32,2047,-226
4700 DATA 2047,-1,-1,-28799,-1,-1,-57,-8065
4710 DATA -1,-1,-7176,8191,-1,-15,-505,-1
```

```
4720 DATA -1,-897,-32257,-1,-1,8176,16383,-1
4730 DATA -57,-505,-1,-1,-897,-16129,-1,-1
4740 DATA -7170,2047,-1,-1,0,16383,-1,-8
4750 DATA 1,-1,-1,-64,15,-1,0,0
4760 '  ｸﾗｳﾝ
4770 DATA -1,-13,-1,-1,-1,-1,-1,-1
4780 DATA -1,-1,-1,-1,-1,16383,-1,-1
4790 DATA -13,-1,-1,-1,-1,-1,-1,-1
4800 DATA -3073,-1,-1,-1,16383,-1,-1,-13
4810 DATA -1,-1,-1017,-193,-2033,-2,3,-16
4820 DATA 31,-128,63,16128,127,-4096,4083,-1024
4830 DATA 1023,3,-1,-4096,16368,127,-1,-32765
4840 DATA -256,4095,-4,63,-4096,-3585,-7232,1023
4850 DATA -32753,-1,-1024,32760,255,-1,-16377,-64
4860 DATA 2047,-8,255,-512,32371,-24704,8191,-4093
4870 DATA -1,-4093,-13,-50,29596,-7,-193,-793
4880 DATA 14799,-97,0,12,0,0,0,480
4890 DATA 0,0,0,3072,0,0,7,-2048
4900 DATA 0,0,127,-32768,0,0,480,0
4910 DATA 0,0,7680,0,0,1,8192,0
4920 DATA 63,-32750,127,0,8191,-15376,-2,15
4930 DATA -1,-16129,-4,511,-16,1023,-32,16383
4940 DATA -1024,4095,-253,-1,0,16383,-4033,-23
4950 DATA 3073,-1,1023,-1023,-8177,-16,16383,-12673
4960 DATA -25345,-255,-4,480,4095,-8161,-64,3072
4970 DATA -2,255,-512,31,-64,2047,-7796,25087
4980 DATA -2048,16383,0,16383,63,-1538,32671,-6145
4990 DATA -31745,-24601,-1538,32760,0,0,0,0
5000 DATA 0,480,0,0,0,3072,0,0
5010 DATA 7,-2048,0,0,127,-32768,0,0
5020 DATA 480,0,0,0,4608,0,0,1
5030 DATA -8192,0,63,-32738,127,0,7175,-15568
5040 DATA -2034,14,3,-16144,28,384,48,-15616
5050 DATA 96,12288,3084,3072,771,3,0,12288
5060 DATA 12336,96,3073,-32765,768,3073,-8180,48
5070 DATA 12288,-12673,-25408,769,-32756,480,3072,24600
5080 DATA 192,3072,-16378,192,1536,24,192,1536
5090 DATA 24588,384,6144,12291,0,12291,51,16383
5100 DATA -3073,-199,-31949,-1,16383,-3176,0,0
5110 '
5120 DATA 512,32,160,64,512,64,80,96
5130 DATA 160,96,80,128,160,128,160,160
5140 '  ﾊｲﾄｳﾋｮｳ ﾋｮｳｼﾞ ﾉ ﾃﾞｰﾀ
5150 DATA 5,-2000,500,9,-1000,500,13,-500,700
5160 DATA 17,200,1000,21,500,2000
```

さくいん

和文

英文

MEMO

FM77AV Visual入門

昭和61年6月20日　初　版

著　者　岳田陽介（たけだようすけ）

発行者　笠原洪平

発行所　工学図書株式会社

（営業所）　東京都千代田区九段北 1-14-15
電話東京（262）3772番
振替東京　7-13465番

印刷所　㈱サンヨー 03（294）4951

ISBN 4-7692-0153-2 C3058　**定価 1,300円**

ISBN4-7692-0153-2 C3058 ¥1300E

定価1,300円

